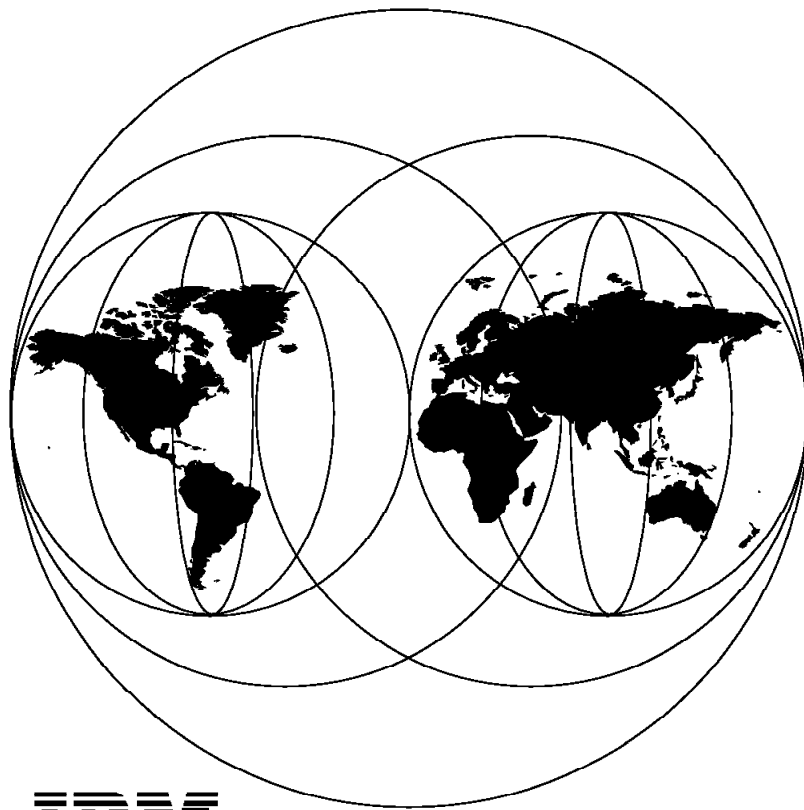


International Technical Support Organization

GG24-4412-00

NetView for OS/2 as an SNMP Manager

December 1994



IBM

**International Technical Support Organization
Raleigh Center**



International Technical Support Organization

GG24-4412-00

NetView for OS/2 as an SNMP Manager

December 1994

Take Note!

Before using this information and the product it supports, be sure to read the general information under "Special Notices" on page xvii.

First Edition (December 1994)

This edition applies to Version 2.0 of NetView for OS/2, Program Number 5871-AAA for use with the OS/2 V2.0 with Service Pak 2, or OS/2 V2.1 with Service Pak 1 licensed programs.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

An ITSO Technical Bulletin Evaluation Form for reader's feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. 985 Building 657
P.O. Box 12195
Research Triangle Park, NC 27709-2195

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1994. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Abstract

This document is unique in its detailed coverage of NetView for OS/2 as an SNMP manager. This new version of the product focuses on SNMP management instead of CMIP management. In addition to managing SNMP-capable devices on the LAN, through the use of proxy agents, and exchanges with other SNMP managers, NetView for OS/2 can help manage the enterprise.

This document was written for systems programmers and LAN administrators, who will be responsible for SNMP management on the Intel platform. Some knowledge of NetView for OS/2 and SNMP is assumed.

(302 pages)

Contents

Abstract	iii
Special Notices	xvii
Preface	xix
How This Document is Organized	xix
Related Publications	xx
International Technical Support Organization Publications	xx
Acknowledgments	xxi
 Chapter 1. Network Protocols and SNMP	 1
1.1 Internet Protocol (IP)	1
1.2 Internetwork Packet Exchange (IPX)	1
1.3 Network Basic Input/Output System (NetBIOS)	1
1.4 AnyNet/2	2
1.5 SNMP Management	2
1.6 Simple Network Management Protocol (SNMP)	3
1.7 What Does an SNMP Agent Do?	4
1.8 What Does an SNMP Proxy Agent Do?	4
1.9 Events and Traps	5
1.10 SNMP Commands	6
1.10.1 SNMPGET	6
1.10.2 SNMPNEXT	7
1.10.3 SNMPSET	7
1.10.4 SNMPTRAP	7
1.10.5 SNMPWALK	8
 Chapter 2. ITSO Lab Environment	 9
2.1 ITSO Lab Environment Hardware	9
2.2 ITSO Lab Environment Software	9
2.3 ITSO Lab Architecture Drawing	10
2.3.1 Software Installed	11
2.4 Installing the NetWare Requester	12
2.5 IBM TCP/IP for OS/2 and Database Manager 2/2 Installation and Configuration	20
2.5.1 Installing TCP/IP Protocol in LAPS	20
2.5.2 Installing TCP/IP	23
2.5.3 OS/2 Database Manager 2/2 Installation	31
2.6 Installing and Configuring NetView for OS/2 Managed Systems	34
2.6.1 Configuring the NetView for OS/2 Agent on a Managed System	34
2.6.2 Implementing Security on the Managed System	36
2.6.3 Installing and Configuring NetView for OS/2 Managing Systems	40
2.7 Starting Sequence for All Management Software	43
2.7.1 Starting a NetView for OS/2 Managed System with No LMU	43
2.7.2 Starting a NetView for OS/2 Managing System with No LMU	43
2.7.3 Starting a NetView for OS/2 Managed System with the LMU Proxy Agent	43
2.7.4 Starting a NetView for OS/2 Managing System with the LMU Proxy Agent	43
 Chapter 3. NetView for OS/2 Functional Overview	 45

3.1	Discovery Processes	45
3.1.1	Description of Discovery Process	45
3.1.2	Using Log Files and Seed Files to Limit Discovery	46
3.1.3	How to Set Up a Management Collection	50
3.2	MIB Loader	60
3.2.1	Obtaining the CISCO Router MIB via FTP	60
3.2.2	Loading the CISCO Router MIB	61
3.2.3	Acquiring MIBs from Other Devices	61
3.3	MIB Browser	62
3.3.1	Browsing the CISCO Router MIB	62
3.3.2	Graphing MIB Variables through the Browser	64
3.4	Building and Executing MIB Applications	66
3.4.1	Memory/Disk Storage Size/Used	67
3.4.2	Graphing CPU Utilization	70
3.4.3	Registering MIB Applications	74
3.4.4	Running Your Applications on a Group of Resources	74
3.5	Data Collector	75
3.6	Event Automation and the Event Displayer	83
3.6.1	Setting Up the Event Automator	83
3.6.2	Using the Event Displayer	88
3.7	Host Connectivity	93
Chapter 4.	OS/2 Agents	95
4.1	SNMP Agent	95
4.1.1	Storage	95
4.1.2	Configuration	99
4.1.3	Printers	101
4.1.4	Software	103
4.2	LMU MIB	106
4.3	The System Information Agent (SIA) Subagent	117
4.3.1	System Shutdown/Reboot	117
4.3.2	Processor Utilization	119
4.3.3	Commands	123
Chapter 5.	LAN Server and LAN Requester Agents	129
5.1	LAN Server Agent	129
5.1.1	General Information	129
5.1.2	Configuration	131
5.1.3	Statistics	134
5.1.4	Performance	136
5.2	LAN Requester Agent	137
5.2.1	General Information	137
5.2.2	Configuration	138
5.2.3	Statistics	139
5.2.4	Commands	141
5.2.5	Performance	142
Chapter 6.	DOS and DOS/Windows Agents	145
6.1	DOS Agents	145
6.2	DOS/Windows Agent	147
Chapter 7.	NetWare Agents	149
7.1	LMU Agents for NetWare	149
7.2	Management Desk - LMU NetWare Objects	149
7.3	Management Desk - NetWare Objects	152

Chapter 8. LMU Interactions	155
8.1 LMU Installation and Customization	155
8.1.1 LMU Installation for OS/2 Stations	155
8.1.2 LMU Installation for DOS Stations	155
8.1.3 LMU Installation for Windows Stations	156
8.1.4 LMU Installation for NetWare Servers	156
8.2 LMU Configuration	157
8.2.1 LMU Configuration for OS/2 Stations	157
8.2.2 LMU Configuration for DOS Stations	164
8.2.3 LMU Configuration for Windows Stations	164
8.2.4 LMU Configuration for NetWare Servers	165
8.3 LMU Startup	166
8.4 LMU Remote Commands from NetView for OS/2	168
8.5 Integrating LMU Alerts into NetView for OS/2	169
8.5.1 Outlining the Systems Management Scenario	169
8.5.2 Setting Up the LMU Managed Station	169
8.5.3 Setting Up the NetView for OS/2 Managing Station	172
8.5.4 Generating and Resolving the Alert Situation	179
8.6 Gathering Vital Product Data (VPD)	182
Chapter 9. Connecting to NetView on MVS/ESA	189
9.1 Connecting to NetView for MVS/ESA	189
9.1.1 Setting Up Communications Manager/2	189
9.1.2 Defining an Implicit Focal Point Relationship	203
9.1.3 Setting Up NetView on the Host	204
9.1.4 Testing Your CM/2 to Host NetView Link	204
9.1.5 Using and Changing Alert Filters	206
9.1.6 Starting the NetView for OS/2 Host Connection Program	208
9.1.7 Using NetView for MVS to Monitor SNMP Devices	211
9.1.8 Initiating Actions from NetView for MVS Using RUNCMD	219
9.1.9 Automating Actions from NetView for MVS	223
Chapter 10. Exploiting REXX Programs and the SNMP Commands	237
10.1 Storing MIB Values in a Database	237
10.2 Collecting and Displaying Combinations of MIB Variable Types	240
10.3 Creating Complex Thresholds and Sending Traps	242
10.4 Automating Changes to a Group of Systems	244
Chapter 11. Scenario 1 - Managing a Lexmark 4039 SNMP Printer	249
11.1 Loading the Lexmark MIB into Our NetView for OS/2 Managing System	249
11.2 Using the MIB Browser to Find Interesting Variables	250
11.3 Setting Up the Data Collector to Monitor Printer Status	253
11.4 Generating an Alert When the Status Changes	257
11.5 Showing the Alert on the Event Displayer	258
Chapter 12. Scenario 2 - Monitoring LAN Server Generic Alerts	261
12.1 Monitoring for Maximum Error Log Size Exceeded	261
12.1.1 Prerequisites for the Remote LAN Server	261
12.1.2 Setting Up the Event Automator	262
12.1.3 Generating the Error at a Remote Server	268
12.1.4 Resolving the Alert Situation	270
12.1.5 IBMLAN.INI File for Remote OS/ LAN Server	270
Appendix A. LMU/2 Related Files	275
A.1 LMU/2 Control File (LMU.CTL) for the Managing System	275

A.2	USERVPD.CFG File	282
A.3	LMU/2 Control File (LMU.CTL) for the Managed System	283
Appendix B. Host NetView Related Files		291
B.1	CM/2 Configuration File	291
B.2	Host VTAM Definitions	292
Appendix C. Configuration Files for Our NV2MGR1 Machine		293
C.1	CONFIG.SYS File	293
C.2	PROTOCOL.INI File	296
Index		299

Figures

1.	SNMP Manager-Agent Model	4
2.	SNMP Proxy Agent Model	5
3.	ITSO Lab Architecture	10
4.	NetWare Installation Window	12
5.	Selecting a Destination Drive for NetWare Files	13
6.	Type of Installation	13
7.	Selecting an ODI Driver	13
8.	Protocol Stacks	14
9.	CONFIG.SYS Statements Added by NetWare	14
10.	CONFIG.SYS after LAPS Has Been Run	15
11.	Selecting an ODI Driver 2	15
12.	IPX Support for DOS and Windows	15
13.	Optional Protocols	16
14.	Save Changes to the CONFIG.SYS	16
15.	Copying All the Drivers on the Disk	17
16.	Copy Requester Files	17
17.	Location of NET.CFG File	17
18.	Editing the NET.CFG	18
19.	NetWare Requester Installed	19
20.	LAPS Installation Initial Screen	20
21.	LAPS Configuration Option	21
22.	LAPS Main Screen	21
23.	Token-Ring Address	22
24.	LAPS Start Update	23
25.	Installation Screen for TCP/IP	24
26.	Icon View of TCP/IP Features	25
27.	TPC/IP Configuration Notebook: Network	25
28.	TCP/IP Routing Configuration	26
29.	Entry Window for Router Information	27
30.	TCP/IP Hostname and Other Services Configuration	27
31.	TCP/IP SNMP MIB-II Information	29
32.	TCP/IP SNMP Traps and PW.SRC File	30
33.	Database Manager 2/2 Installation Initial Screen	31
34.	Database Manager 2/2 Options Installation	32
35.	Database Manager 2/2 Configure Installation	33
36.	Database Manager 2/2 Installation Successful	33
37.	NetView for OS/2 Installation Directories - Agent Code	34
38.	NetView for OS/2 Installation Directories - Agent Documentation	35
39.	NetView for OS/2 Installation Transports for Agents	36
40.	SNMP Community Names for Agents - TCP/IP Environment	37
41.	SNMP Community Names for Agents - NetBIOS Environment	38
42.	SNMP Trap Destinations (IP)	38
43.	SNMP Trap Destinations (NetBIOS)	39
44.	NetView for OS/2 SNMP Configuration for Agents	40
45.	Startup Folder for the NV2MGR1 Machine without NetView for OS/2 Icons	42
46.	NetView for OS/2 Icon View Folder for the NV2MGR1 Machine	42
47.	STARTUP.CMD File for NetView for OS/2 Managing Station with LMU Proxy	44
48.	All Systems Folder with All Systems Discovered	47
49.	All Systems Folder Created from SEED File	50

50.	Example of an Offline System	52
51.	Management Collection Template	52
52.	Management Notebook for Custom Collections	53
53.	Setting the Timed Refresh Interval	54
54.	Creating a New Attribute Filter	54
55.	Entering a Name for the Filter	55
56.	Setting the Filter Attributes	56
57.	Giving the Filter Icon a Name	56
58.	Press Apply	57
59.	Setting the Management Collection Name	57
60.	New Folder of Offline Systems	58
61.	Sample Management Collections	59
62.	Panel Showing Commands Needed to Get CISCO MIB	60
63.	Output from Running LOADMIB Command	61
64.	MIB Browser Window Showing Results from Query	63
65.	Contents of the BROWSER.SAV File	64
66.	Example of MIB Variables Being Graphed (dnReceived and dnHellos)	66
67.	Icon for Application Builder in the Systems Folder	67
68.	MIB Application Builder Screen	68
69.	Successful Build of MIB Application	69
70.	MIB Applications Folder	69
71.	Sample Run of a Custom MIB Application	70
72.	Application Builder Window for CPU Utilization Graph	72
73.	MIB Application Execution - CPU Utilization Graph	73
74.	MIB Registration	74
75.	Data Collection Icon in NetView Folder	75
76.	MIB Data Main Screen	76
77.	Selecting an Object in the Tree	77
78.	Entering a Label Name	78
79.	Setting the Thresholds and Storing Data	79
80.	Data Collector Window	80
81.	Show Data Collected	81
82.	Show Data Graphed	82
83.	NetView for OS/2 Main Icon View	83
84.	Event Automation Update Window	84
85.	Event Automation - Add Enterprise Name and ID	85
86.	Event Automation - LAN Requester Trap Name and ID Added	85
87.	Event Automation - Adding Specific Trap Information	86
88.	Adding Specific Traps and Selecting Actions for These Traps	87
89.	Event Automation - Adding Automation Actions	88
90.	Selecting Event Displayer from the NetView for OS/2 Icon View	89
91.	Event Displayer Window Showing Event Types	89
92.	OS/2 Window Showing Location and Size of Logs	90
93.	Event Displayer - Selecting Only Enterprise-Specific Traps	91
94.	Event Displayer - All Events Window	92
95.	Pop-Up Message Showing Operation State Change of LAN Requester	92
96.	Physical Storage	96
97.	Logical Storage Description	97
98.	Logical Storage Size	98
99.	Logical Storage Usage	99
100.	Device Description	100
101.	Device Errors	101
102.	Printer Status	102
103.	Printer Errors	103
104.	Software Installed	104

105.	Software Running	105
106.	LMU Proxy Variable	106
107.	LMU Profile Variable	107
108.	LMU Query Variable	108
109.	LMU Topology Entry Variable	109
110.	LMU Alert Entry Variable	110
111.	LMU Database Configuration Computer Variable	111
112.	LMU Database Configuration Machine Name Variable	112
113.	LMU Database Configuration Total Memory Variable	113
114.	LMU Database Software Program Name Variable	114
115.	LMU Database Software Current CSD Variable	115
116.	LMU Database Hardware Adapter Name	116
117.	DB2/2 Query Manager	117
118.	Reboot System	118
119.	Shutdown System	119
120.	Interval Description	120
121.	Processor Idle	121
122.	Processor Busy	122
123.	Command Names	124
124.	Command Description	125
125.	Command Definition	126
126.	Command Results	127
127.	Logged On User Table	130
128.	LAN Server Usage State	131
129.	Access Alert	132
130.	Logon Alert	133
131.	Disk Alert	134
132.	Average Server Response Time	135
133.	Failed Attempts to Allocate Big Buffer	136
134.	Server Cache Flushed	137
135.	IBM LAN.INI File Name	138
136.	Work Heuristics	139
137.	Failed Requester Big Buffer Allocation	141
138.	Command Options	142
139.	LAN Requester Poll Performance	143
140.	MIB Browser View of LMU Database	146
141.	Query Manager View of LMU Database	146
142.	LMU DOS/Windows Object	147
143.	LMU Remote Command - Execute Panel	147
144.	LMU Remote Command - Results Panel	148
145.	LMU NetWare Object	149
146.	LMU Display Panel - NetWare Requester	150
147.	LMU Icon - NetWare Server	150
148.	LMU Display Panel - NetWare Server	151
149.	All the Different Icons for Our NV2MGR1 Machine	151
150.	LMU Display Attributes - Showing OS/2 Interoperability	152
151.	NetWare Object	152
152.	NetWare Quick Status - Results	152
153.	NetWare Node Information - Overview	153
154.	NetWare Node Information - LAN Boards	153
155.	IBM TCP/IP for OS/2 Configuration - REXEC Customization Panel	158
156.	IBM TCP/IP for OS/2 TCP/IP Configuration - Autostart Panel	159
157.	IBM TCP/IP for OS/2 Configuration - SNMP TRAPDST Configuration	160
158.	LMUCUST Command Output	163
159.	LMUSTART.CMD File	163

160.	LMU Managing Station Panel Showing Initial Messages	167
161.	LMU Proxy Agent and SNMPD Panel Showing Initial Messages	168
162.	LMUSTART.CMD on the LMU Managed Machine	171
163.	APPWATCH.TAB Table on LMU Managed Machine	172
164.	Event Automation Update Window - Adding LMU Type Alerts	173
165.	Adding Generic/Specific LMU Traps to the Event Automator	174
166.	Specifying Actions Upon Receipt of the LMU Alert	175
167.	MIB Browser Showing the Format of LMU Alerts	176
168.	APPSTART REXX Command File	178
169.	LMU GUI - Managing System and Node View	179
170.	LMU GUI - APPWATCH Alert Has Arrived - Display Events	180
171.	Viewing the Alert from the LMU GUI Display Events Window	180
172.	Using NetView for OS/2 Event Displayer to Show LMU Alerts	181
173.	Event Automator - Default Pop-Up for LMU Traps Received	182
174.	Communications Manager/2 - Icons/Functions Available	190
175.	Communications Manager/2 Setup Window	190
176.	Communications Manager/2 - Open Configuration Window	191
177.	Communications Manager/2 - Configuration Definition for WTRMODEL	192
178.	Communications Manager/2 Profile Listing	193
179.	Setting LAN DLC Adapter Parameters	194
180.	CM/2 Profile List - Selecting SNA Node Characteristics	195
181.	Setting SNA Local Node Characteristics	195
182.	Setting SNA Local Node Options	196
183.	CM/2 Profile List - Selecting SNA Connections	197
184.	Communications Manager/2 Host Connections List	197
185.	SNA Connections - Adapter Selection List	198
186.	SNA Connections - Define Link Names and Destination Address	199
187.	Define the Host NetView Application as Our APPC Partner LU	200
188.	Communications Manager/2 Profile List - Selecting SNA Features	201
189.	CM/2 Configuration - SNA Features List - Local LUs	202
190.	CM/2 Configuration - SNA Features List - Partner LUs	202
191.	CM/2 Configuration - SNA Features List - Modes	203
192.	Initiating Subsystem Management to Test Your Configuration	204
193.	Subsystem Management - All Subsystems Started	205
194.	Selecting LU 6.2 Sessions from SNA Subsystems	205
195.	Successful LU 6.2 Session Establishment	206
196.	Contents of TRALERT.FLT File	207
197.	RuleContent for LAN Server/Requester Filter	207
198.	Contents of TRALERT.CFG File	208
199.	Starting the NetView for OS/2 Host Connection Program	208
200.	Starting TRALERTD with our LAN Server/Requester RuleContent	209
201.	Successful Start of the Host Connection Program Using Defaults	210
202.	SNMP Traps which Will Be Forwarded to Host NetView	210
203.	NetView for MVS - Hardware Monitor Alerts-Dynamic Screen	211
204.	Sample Host NetView Alert Screen Showing SNMP Alerts	212
205.	Host NetView Alerts-Static Screen	213
206.	Host NetView - Recommended Actions for Selected Event Screen	214
207.	Host NetView - Event Detail Display Menu	215
208.	Host NetView - Event Details (Page 1 of 2)	216
209.	Host NetView - Event Details (Page 2 of 2)	217
210.	Host NetView - Product Set Identification	218
211.	Host NetView RUNCMD Syntax	219
212.	Host NetView - Using RUNCMD to Enter OS/2 and LAN Commands	220
213.	LMU Network of Managed Machines	221
214.	Using LMU LMUCMD to Issue Commands to Remote Stations	222

215. LAN Requester Alert on Host NetView Console	224
216. Host NetView Recommended Action for LAN Requester Alert	225
217. Host NetView - Event Detail Menu for LAN Requester Alert	226
218. Event Details for LAN Requester Alert	227
219. Pseudo Code of our Automation Table Entry	228
220. Section of Host NetView AUTOTBL for LAN Alerts	229
221. Selecting to See Hexadecimal Display of Subvectors	230
222. Hexadecimal Display of Data Record (Page 1 of 2)	231
223. Hexadecimal Display of Data Record (Page 2 of 2)	232
224. Stopping OS/2 LAN Requester and Host NetView Restarting It	233
225. Automation Table Entry Highlights LAN Requester Alert	234
226. Command List Called From our Automation Table Entry	234
227. STRTREQ CLIST Running on WTWKSH2 NetView Operator Console	235
228. LMUPOPUP Message Stating Requester Has Been Restarted	235
229. Query of TESTTBL Table in Query Manager	240
230. Multiple Types of MIB Values	242
231. Pop-Up Message for Exceeded Disk Capacity	244
232. Value of Syslocation before Changes	245
233. SYSLOCATION after the REXX Program Has Executed	247
234. MIB Browser Showing Enterprise-Specific MIBs	250
235. 4039 Status MIB Variables	251
236. Status of InputEmpty MIB Variable	253
237. Adding the InputEmpty MIB Variable to Data Collector	254
238. Setting the Polling Interval and Thresholds on InputEmpty	255
239. Collecting InputEmpty Data Values	256
240. Showing InputEmpty Data Collected Over Time	257
241. Graph Showing Where StatusInputEmpty Generated Alert	258
242. Event Displayer Showing StatusInputEmpty Alert	259
243. NetView for OS/2 Main Icon View	262
244. Event Automation Update Window	263
245. Event Automation - Add Enterprise Name and ID	264
246. Event Automation - Adding Specific Trap Information	265
247. Event Automation - Adding the Istrap32MaxErr Trap	266
248. Event Automation - Adding the Automation Actions	267
249. LMU Network Showing Managing and Managed Stations	268
250. Event Displayer Window Showing LAN Server Alert #32	269
251. Event Automation - Pop-Up Showing Trap Received: Istrap32MaxErr	269
252. FIXERRLG REXX Command File to Resolve MAXERRORLOG Alert	270

Tables

1. Possible Event Types and Descriptions	90
2. LAN Server and LAN Requester CLASS Identifiers	207
3. Cause Undetermined (X'97') Subvector Code Points and Text	228
4. Detailed Data (X'98') Subvector Data ID, Data Types and Text	229
5. IBM 4039-16L LaserPrinter MIB Status Variables	252

Special Notices

This publication is intended to help LAN administrators who are responsible for network management on the Intel platform. It will help them install the product, and provide some sample configuration parameters that can be used to help manage the LAN. The information in this publication is not intended as the specification of any programming interfaces that are provided by any other organizations. See the PUBLICATIONS section of the IBM Programming Announcement for NetView for OS/2 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (VENDOR) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

You can reproduce a page in this document as a transparency, if that page has the copyright notice on it. The copyright notice must appear on each page being reproduced.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

Advanced Function Printing	AFP
AIX	AnyNet
AT	DATABASE 2
DB2/2	FFST/2
First Failure Support Technology/2	IBM
Library Reader	Micro Channel
MVS/ESA	NetView
Operating System/2	OS/2
Personal System/2	Presentation Manager
PS/2	RISC System/6000
RMONitor	RS/6000
Trouble Ticket	XGA

The following terms are trademarks of other companies:

Apple, Macintosh	Apple Computer, Incorporated
Cisco	Cisco Systems, Incorporated
DEC, DECnet	Digital Equipment Corporation
Frame	Frame Technology, Incorporated
Intel, 386, 486, 80386	Intel, Incorporated
Lexmark	Lexmark International, Incorporated
Microsoft, Windows	Microsoft Corporation
Novell, NetWare, IPX	Novell, Incorporated
Network File System, NFS	Sun Microsystems, Incorporated

Other trademarks are trademarks of their respective companies.

Preface

This document provides early experience with NetView for OS/2. In addition to showing how the specific functions of the product work, the scenarios in this book show how the product works with other products to provide solutions.

It contains a description of functions in NetView for OS/2 the functions of agents and examples of how to manage DOS, WIN OS/2 and NetWare and some sample scenarios.

How This Document is Organized

The document is organized as follows:

- Chapter 1, "Network Protocols and SNMP"

This chapter provides an overview of the protocols that NetView for OS/2 uses for managing its environment. In addition, an overview of SNMP V1 is provided, with some examples of how to issue some SNMP commands.

- Chapter 2, "ITSO Lab Environment"

This chapter shows a sample environment that is going to be managed in the scenarios later in the book. In addition it shows how to install the prerequisite products and what tailoring is required to have them work with NetView for OS/2.

- Chapter 3, "NetView for OS/2 Functional Overview"

This chapter provides a functional overview of NetView for OS/2. Examples of how to set up and use many of the functions relating to discovery, and managing MIB variables are provided.

- Chapter 4, "OS/2 Agents"

This chapter shows how the NetView for OS/2 agent interacts with the NetView for OS/2 manager.

- Chapter 5, "LAN Server and LAN Requester Agents"

This chapter shows what the LAN Server and LAN Requester agents can provide NetView for OS/2 as a manager of these resources.

- Chapter 6, "DOS and DOS/Windows Agents"

This chapter shows how to set up and use the DOS and DOS/Windows agents from NetView for OS/2.

- Chapter 7, "NetWare Agents"

This chapter shows how to install, set up and use the Novell NetWare agents in a NetView for OS/2 environment.

- Chapter 8, "LMU Interactions"

This chapter show how LMU is integrated into the NetView for OS/2 management environment. In addition, examples are provided of how to install and tailor LMU for interactions with OS/2, DOS, Windows and NetWare.

- Chapter 9, "Connecting to NetView on MVS/ESA"

This chapter shows how to exchange information with other management platforms. Samples of how to set up the alert flow with NetView running on MVS/ESA are provided.

- Chapter 10, “Exploiting REXX Programs and the SNMP Commands”

This chapter shows some samples of how to use REXX programs to extract management information, and provide some simple, but useful reports.

- Chapter 11, “Scenario 1 - Managing a Lexmark 4039 SNMP Printer”

This chapter shows how to manage a Lexmark SNMP printer from NetView for OS/2.

- Chapter 12, “Scenario 2 - Monitoring LAN Server Generic Alerts”

This chapter shows how to use the LAN Server agent and how to set up and handle alerts from the agent.

- Appendix A, “LMU/2 Related Files”

This appendix provides samples of the control files that are used to tailor LMU.

- Appendix B, “Host NetView Related Files”

This chapter provides samples of how to customize Communications Manager/2 and NetView for MVS, so alerts can be sent from NetView for OS/2.

- Appendix C, “Configuration Files for Our NV2MGR1 Machine”

This chapter provides samples of OS/2-related configuration files.

Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this document.

- *IBM NetView for OS/2 Developer's Guide*, SC31-8098
- *IBM NetView for OS/2 User's Guide*, SC31-8099
- *IBM NetView for OS/2 Installation and Administration Guide*, SC31-8100
- *IBM NetView for OS/2 Agents*, SC31-8102
- *NetView Automation Implementation*, LY43-0015 (available to IBM licensed customers only)

International Technical Support Organization Publications

A complete list of International Technical Support Organization publications, with a brief description of each, may be found in:

International Technical Support Organization Bibliography of Redbooks, GG24-3070.

To get a catalog of ITSO technical bulletins (redbooks) online, VNET users may type:

TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG

How to Order ITSO Technical Bulletins (Redbooks)

IBM employees in the USA may order ITSO books and CD-ROMs using PUBORDER. Customers in the USA may order by calling 1-800-879-2755 or by faxing 1-800-284-4721. Visa and Master Cards are accepted. Outside the USA, customers should contact their IBM branch office.

Customers may order hardcopy redbooks individually or in customized sets, called GBOFs, which relate to specific functions of interest. IBM employees and customers may also order redbooks in online format on CD-ROM collections, which contain the redbooks for multiple products.

Acknowledgments

The advisor for this project was:

Barry D. Nusbaum

International Technical Support Organization, Raleigh Center

The authors of this document are:

John Barker IBM UK

Dan Heimann IBM Canada

Miroslav Iwachow IBM Czech Republic

This publication is the result of a residency conducted at the International Technical Support Organization, Raleigh Center.

Thanks to the following people for the invaluable advice and guidance provided in the production of this document:

Rob Macgregor, Dave Shogren

International Technical Support Organization, Raleigh Center

Dave Fresquez, Jeanette Lee, Roger Rea, Colleen Miles, Jim Chou
IBM RTP Development

Request for Feedback

Readers of this document are encouraged to feed back any information or comments regarding *any* of the material in this document. Please send your comments to:

Barry D. Nusbaum
ITSO-Raleigh
VNET: BARRY at WTSCPOK

or: IBM Corporation 545/B657/BB106
Attn: Barry D. Nusbaum
Building 657 Rm BB106
4912 Green Road
Raleigh NC 27604

INTERNET: bnusbaum@vnet.ibm.com

Chapter 1. Network Protocols and SNMP

NetView for OS/2 provides a management system for administering, controlling, and monitoring numerous devices that make up multi-vendor, heterogeneous LANs and WANs. In order to achieve this aim, NetView for OS2 supports a number of network transports that are common in PC workgroup environments. These are:

1. IP
2. IPX
3. NetBIOS
4. SNA (using AnyNet/2 Sockets over SNA)

The following sections describe these protocols and their functions.

1.1 Internet Protocol (IP)

Transmission Control Protocol/Internet Protocol (TCP/IP) is a suite of protocols that address multi-vendor interoperability. These protocols define applications, transport controls, networking, routing, and network management. TCP/IP is defined as having three layers. These are the application layer, the transport layer, and the network layer. IP belongs to the network layer and provides a connectionless delivery service on this layer by creating packets for transmission from the source host to the next node in the network. It adds no reliability, flow control, or error recovery, so packets or datagrams sent by IP may be lost, out of order, or even duplicated. It is left to the higher layers to provide these functions. IP is also responsible for connecting the different networks to form one large network.

1.2 Internetwork Packet Exchange (IPX)

The Novell IPX protocol provides a best effort data packet delivery service but does not guarantee the delivery of the data. IPX is a connectionless, full-duplex protocol; that is, it transmits data to a remote node, but does not wait for a response or acknowledgment indicating that the data has been received successfully. It is left to the higher level protocols to provide this guaranteed data transmission. In the NetWare environment, this is implemented within the NetWare shell. As the IPX protocol is connectionless and does not wait for acknowledgment, it is able to provide much higher performance than a connection, or session-based protocol such as NetBIOS.

1.3 Network Basic Input/Output System (NetBIOS)

A large number of applications use NetBIOS for communications on workstations connected to a LAN. The NetBIOS rules do not specify what hardware, software, protocols or physical media are to be used. Therefore, NetBIOS can be implemented on other platforms. NetBIOS was developed initially by IBM for its own PCs on IBM PC networks, but has since been adopted by many other vendors on other platforms.

The popularity of NetBIOS is due to several factors:

1. The NetBIOS interface isolates the application from the communications process.
2. NetBIOS interface calls are independent of the physical hardware and of the communications protocol.
3. The commands for token-ring are the same as for Ethernet or PC Network.
4. The commands for TCP/IP are the same as for other protocols.
5. The application program only concerns itself with the NetBIOS interface and lets NetBIOS perform the communications tasks.

NetBIOS applications are easily ported from one system to another. For example, an application developed on a PC can be moved to a more powerful workstation which uses a different communications protocol. The NetBIOS support programs in each system take care of the differences. NetBIOS is also popular because it makes program-to-program communication easier by hiding the complexities of communications to the user.

Only one NetBIOS unique name can be active on a NetBIOS network; however, no permanent relationship exists between a name and a specific workstation. Once a user is finished using a name, another workstation can use it.

The main tasks of the NetBIOS protocol are to:

1. Find a name on a network
2. Establish and maintain a session between two names
3. Handle error conditions

Network Basic Input Output System (NetBIOS) is a well-accepted proprietary higher-level LAN protocol, providing a session-level programming interface.

NetBIOS supports a layered communications architecture. Names rather than network addresses are used by NetBIOS applications for session support. NetBIOS support provides services to locate and associate these names with MAC network addresses.

1.4 AnyNet/2

NetView for OS/2 uses AnyNet/2 V2.0 Sockets over SNA to connect isolated TCP/IP networks across an SNA network.

Sockets over SNA provides a sockets application programming interface (API) for OS/2. This API presents a TCP/IP sockets interface using the internet address family to connect applications and uses APPC as the underlying transport. This allows the transparent flow of SNMP management information from a remote TCP/IP network via an SNA network to NetView for OS/2.

1.5 SNMP Management

SNMP is a transaction-oriented protocol that allows network elements to be queried directly. It is a simple protocol that allows management information for a network element to be inspected or altered by a system administrator at a network management station.

NetView for OS/2 uses SNMP to manage TCP/IP, IPX, NetBIOS, and SNA/AnyNet networks. A multi-protocol SNMP daemon that provides support for all of these protocols is supplied as part of the NetView for OS/2 package.

1.6 Simple Network Management Protocol (SNMP)

SNMP is a TCP/IP network management protocol and is based on a manager-agent interaction. The SNMP manager (such as NetView for OS/2) communicates with its agents. Agents gather management data and store it, while managers solicit this data and process it.

The SNMP architectural model is a collection of network management stations and network elements, such as gateways, routers, bridges and hosts. These elements act as servers and contain management agents which perform the network management functions requested by the network management stations. The network management stations act as clients; they run the management applications which monitor and control network elements. SNMP provides a means of communicating information about network resources between the network management stations and the agents in the network. This information can be status information, counters, identifiers, and more. The limit is not on the format of the data, but on understanding the value of the data.

The SNMP manager continuously polls the agents for error and statistical data. The performance of the network will be dependent upon what the polling interval is set at. The physical and logical characteristics of network objects make up a collection of information called a Management Information Base (MIB). The individual pieces of information that comprise a MIB are called MIB objects and they reside on the agent system. These MIB objects can be accessed and changed by the agent at the manager's request. This is how NetView for OS/2 manages network objects. Other SNMP managers can also access these MIBs.

NetView for OS/2 supports the following MIBs:

- Standard MIB

This is a standard definition which defines the data layout (length of fields, what the field is to contain) for the management data of a resource. The standard MIB object definitions, MIB-I and MIB-II, enable the monitoring and control of SNMP managed devices. Agents contain the intelligence required to access these MIB values.

- Enterprise-specific MIB

These MIBs are usually unique and proprietary in nature, but are required to support a standard set of common managed object definitions. This ensures that the information they contain can be accessed and modified by agents.

NetView for OS/2 provides the ability to load enterprise-specific MIBs from a MIB description file. By loading this description file you monitor and control vendor devices. NetView for OS/2 supplies enterprise-specific MIBs using the System Information Agent (SIA), LAN Server and LAN Requester SNMP subagents. These unique extensions to the MIB provide applications with the capability to provide the manager with a lot of information. The fact that you can use a command to load new MIBs provides the potential for automation and lots of flexibility. It may be possible to discover new devices, and if there is a new MIB for that device (or application), to dynamically load it and provide management capability for it.

1.7 What Does an SNMP Agent Do?

The SNMP agent is responsible for managed resources and keeps data about the resources in a MIB. The SNMP agent has two responsibilities:

1. To place error and statistical data into the MIB fields
2. To react to changes in certain fields made by the manager

In summary the following steps describe the interactions that take place in an SNMP-managed network:

- Agents store vital information about their respective devices and networks. This information is stored in a MIB.
- The SNMP manager polls each agent for MIB information and stores and displays this information at the SNMP manager station. In this manner the system administrator can manage the entire network from one management station.
- Agents also have the ability to send unsolicited data to the SNMP manager. This is called a trap. A trap is generally a network condition detected by an SNMP agent that requires immediate attention by the system administrator. In SNMP V1 there is no guarantee that the trap will be delivered to the manager. It may be that the agent needs some intelligence to determine if no action has been taken after a period of time, to retransmit the trap. This could be from some logic within the agent to see if the manager has responded within a set period of time.

MANAGER AGENT MODEL

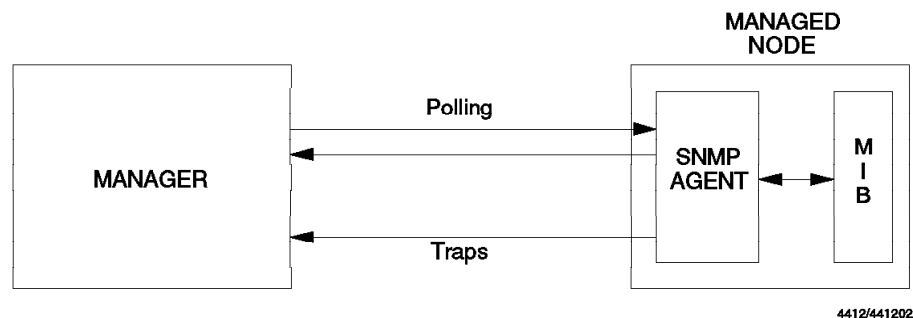


Figure 1. SNMP Manager-Agent Model

1.8 What Does an SNMP Proxy Agent Do?

An SNMP proxy agent has two functions:

1. To perform a subset of the SNMP function, to reduce the processing load on the SNMP manager itself.
2. To act as a management station for devices and networks that cannot be directly managed by the SNMP manager.

NetView for OS/2 uses the second function to enable management of DOS and DOS/Windows workstations running LMU agents.

Information available from the proxy agent about the LMU environment is contained in the LMU management information block (MIB), and includes the following kinds of information:

- Information about the proxy agent workstation itself (name, node address, profile (LMU.INI) data, which LMU programs are running, and more)
- Status information about LMU managing and managed systems
- Alert information from LMU managed systems
- Configuration and performance information from the LMU database

A complete definition of the LMU MIB is supplied in the file LMUMIB. SCR.

Changes in status information about LMU managing and managed systems are converted by the SNMP proxy agent to SNMP traps. LMU alerts that are forwarded to the SNMP proxy agent from the fault manager are converted to SNMP traps.

PROXY AGENT MODEL

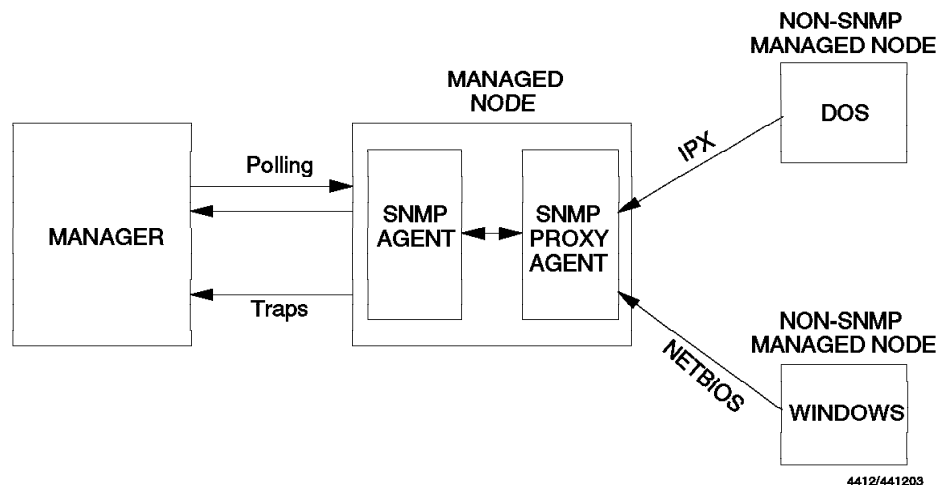


Figure 2. SNMP Proxy Agent Model

1.9 Events and Traps

In a network managed with SNMP, network events are called *traps*. A trap is a message sent from an SNMP agent to a manager without a specific request from the manager.

SNMP defines six generic types of traps and allows the definition of enterprise-specific traps. This trap structure provides the following information to NetView for OS/2:

- The particular agent object that was affected
- Event description (including trap number)
- Time stamp
- Optional enterprise-specific trap identification

- List of variables describing the trap

1.10 SNMP Commands

The SNMP protocol provides commands that managers can use to communicate with agents about network objects. The five commands that can be used are:

- GET

Requests that the responding agent supply values for one or more MIB variables defined to a particular network object. NetView for OS/2 has implemented SNMP V1, so it can't do a get-bulk function.

- SET

Writes new data to one or more MIB variables defined for a network object. You might, for example, need to update information about the network whose description was provided to you by the GET request. In order to SET a value, you will need write access to the MIB variable. A prerequisite to that would be having the same community name specified in both the manager and the agent request. It is similar to specifying a password. In SNMP V1 this type of security is very simplistic.

- GET NEXT

Requests information for the next variable of the object managed by the agent. You can use GET NEXT to get all the variables down the chain until you reach the end.

- TRAP

Sends an unsolicited alert to the SNMP manager. This mechanism can be used by an agent to send status information, or other types of information. It does not have to come from a device. An application might send a trap to indicate a status change, or to trigger an event in automation.

- GET RESPONSE

Contains the data that has come from an agent in response to a get request.

NetView for OS/2 provides five commands that allow use of these commands, either from the OS/2 command line or from within REXX EXECs. For examples of REXX programs using these commands see the following 5 sections. To monitor traps you can issue the \TCPIP\BIN\SNMPTRAP.EXE command which provides a PM interface to monitor traps being sent to a system.

Note: Do not confuse the SNMPTRAP.EXE file with the SNMPTRAP.CMD file in \anv2\bin directory.

1.10.1 SNMPGET

The SNMPGET command uses the SNMP Get Request to query a node object for information. Up to 20 fully-qualified object identifiers can be used as arguments. An example of the use of this command could be to find out the system description and up-time for the node *nvclient*.

The format of the command would be as follows:

```
snmpget nvclient system.sysDescr.0 system.sysUptime.0
```

The following shows the output from the command:

```
system.sysDescr.0 : DISPLAY STRING- (ascii): NetView for OS/2 base SNMP V2
system.sysUpTime.0 : Timeticks: (10769200) 1 day, 5:54:52.00
```

1.10.2 SNMPNEXT

The SNMPNEXT command uses the SNMP Get Next request to query a node for information. For each variable queried, the content of the next value down the tree is returned.

We will use the same argument as for the previous example, only this time we will query the *system* variable.

The format of the command would be as follows:

```
snmpnext nvclient system.sysDescr.0
```

The following shows the output from the command:

```
system.sysObjectID.0 : OBJECT IDENTIFIER: .iso.org.dod.internet.
private.enterprises.ibm.ibmProd.61
```

1.10.3 SNMPSET

The SNMPSET command issues and SNMP Set Request to alter MIB objects and return the result. Again, up to 20 fully-qualified object identifiers can be used as arguments.

In the following example, the system contact name for node *nvclient* is set to Ben Franklin.

The format of the command would be as follows:

```
snmpset nvclient system.sysContact.0 octetstring "Ben Franklin"
```

The following shows the output from the command:

```
system.sysContact.0 : DISPLAY STRING- (ascii): BEN FRANKLIN
```

1.10.4 SNMPTRAP

The SNMPTRAP command issues an SNMP trap to a node.

The following example sends an SNMP trap from node *barry* to managing system *nv2mgr1* with a time stamp equal to zero. The generic-trap is set to 6, while the specific-trap is set to 1. The information passed with the trap is from the variable *system.sysDescr.0*.

The `snmptrap` command was issued from an AIX RISC System/6000. If you enter `snmptrap` from OS/2, it will bring up a Presentation Manager window showing you what traps have been received by OS/2 TCP/IP.

The format of the command would be as follows:

```
snmptrap nv2mgr1 "" barry 6 1 0 system.sysDescr.0 octetstring "OS/2 2.11 machine"
```

1.10.5 SNMPWALK

The `SNMPWALK` command uses the SNMP Get Next request to query node information. Unlike the `SNMPNEXT` command, `SNMPWALK` keeps returning values until the tree structure is exhausted.

The following is the command to request the *system* subtree for node *nvclient*.

```
snmpwalk nvclient system
```

The following shows the output from the command:

```
system.sysDescr.0 : DISPLAY STRING- (ascii): NetView for OS/2 base SNMP V2
system.sysObjectID.0 : OBJECT IDENTIFIER: .iso.org.dod.internet.
private.enterprises.ibm.ibmProd.61
system.sysUpTime.0 : Timeticks: (10873300) 1 day, 6:12:13.00
system.sysContact.0 : DISPLAY STRING- (ascii): Ben Franklin
system.sysName.0 : DISPLAY STRING- (ascii): nvclient.itso.ral.ibm.com
system.sysLocation.0 : DISPLAY STRING- (ascii): Raleigh ITSC Bldg 657
system.sysServices.0 : INTEGER: 72
```

Chapter 2. ITSO Lab Environment

2.1 ITSO Lab Environment Hardware

- PS/2 486 Model 95 Manager and Managed #1
- PS/2 486 Model 95 Manager and Managed #2
- PS/2 386 Managed (OS/2 agents)
- PS/2 386 DOS and DOS/Windows

2.2 ITSO Lab Environment Software

This chapter will describe what each product does, and how it fits into the environment. In addition, the installation of the products will be documented.

Other equipment and systems that were involved with this project were:

- NetWare
- LAPS
- TCP/IP
- DB2/2
- NetView on MVS/ESA
- AIX NetView/6000
- NetView for Windows
- OS/2
- LMU
- AIX NetView Service Point
- OS/2 TCP/IP
- AnyNet/2 (TCP/IP over SNA)
- CM/2
- DOS, and DOS/Windows
- LAN Server, LAN Requester

2.3 ITSO Lab Architecture Drawing

Following is a graphic of our ITSO lab environment showing our managing and managed systems, plus how we interface into the rest of the ITSO network as shown in Figure 3.

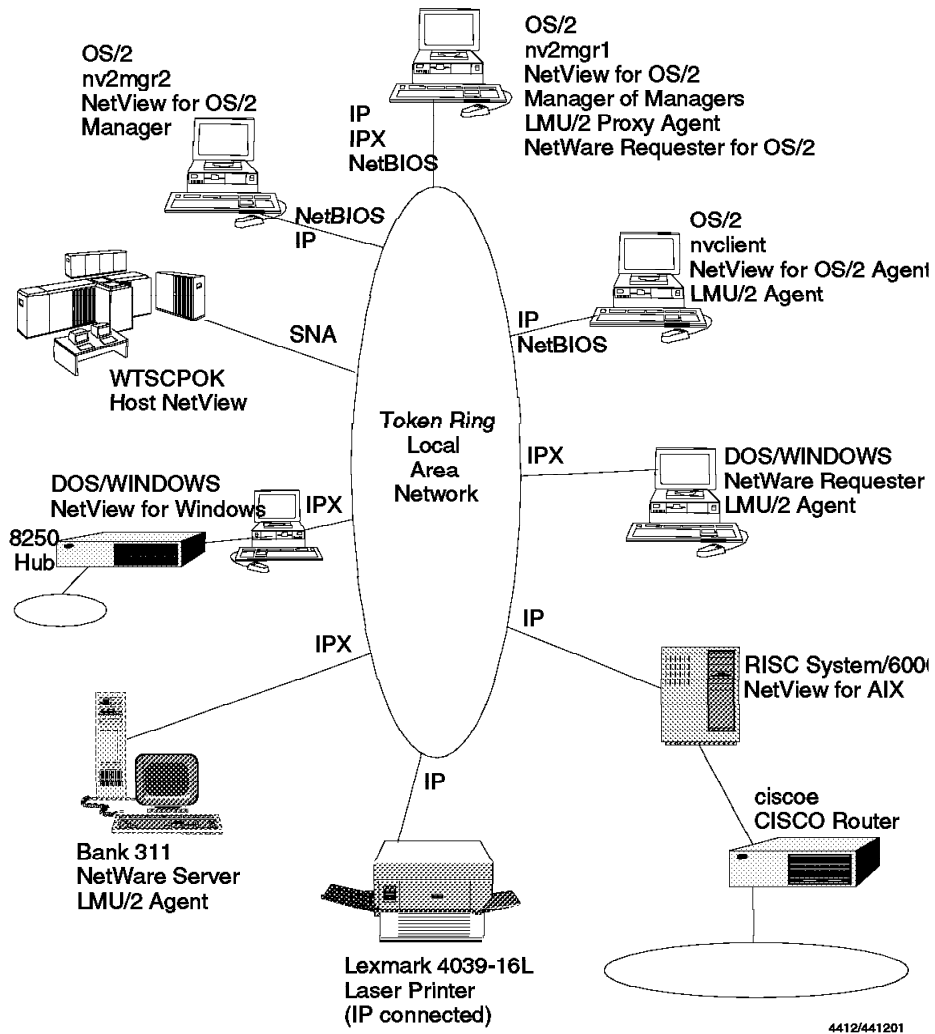


Figure 3. ITSO Lab Architecture

2.3.1 Software Installed

The following table shows the software installation on each of the machines used in the lab.

NV2MGR1	NV2MGR2	NVCLIENT	CSD LEVEL
OS/2 2.11	OS/2 2.11	OS/2 2.11	XR06200
CM/2 1.11	CM/2 1.11	CM/2 1.11	WR06150
DB2/2 1.0	-	-	WR07025
LR 3.0	LR 3.0	LR 3.0	IP07045
	LS 3.0	-	IP07045
TCP/IP 2.0	TCP/IP 2.0	TCP/IP 2.0	UN50382
AnyNet/2 2.0	-	-	WR00000
NetWare Req.	-	-	
LAPS	LAPS	LAPS	WR07045
LMU/2 Manager	LMU/2 Client	LMU/2 Client	LM00201
NetView for OS/2	NetView for OS/2	NetView for OS/2	IC01000
-Managing	-Managing	Agents Only	

DOSWCLI

Windows 3.1
LMU/2 Windows
Client

2.4 Installing the NetWare Requester

To get access to the Novell NetWare 3.11 and 4.01 servers we installed the NetWare for OS/2 requester. For a description of installing the LMU code on the NetWare server see 8.1, "LMU Installation and Customization" on page 155 for NetWare servers. Note that LAN Server and the NetWare for OS/2 requester cannot be loaded on the same machine. This section describes a simple installation of the NetWare requester.

1. Start installation by placing NetWare for OS/2 Requester disk 1, marked WSOS21 into drive A:.
2. Type install.
3. In the NetWare Workstation for OS/2 Installation window click on **Installation** in the menu bar at the top and select **Requester on Workstation** as shown in Figure 4.

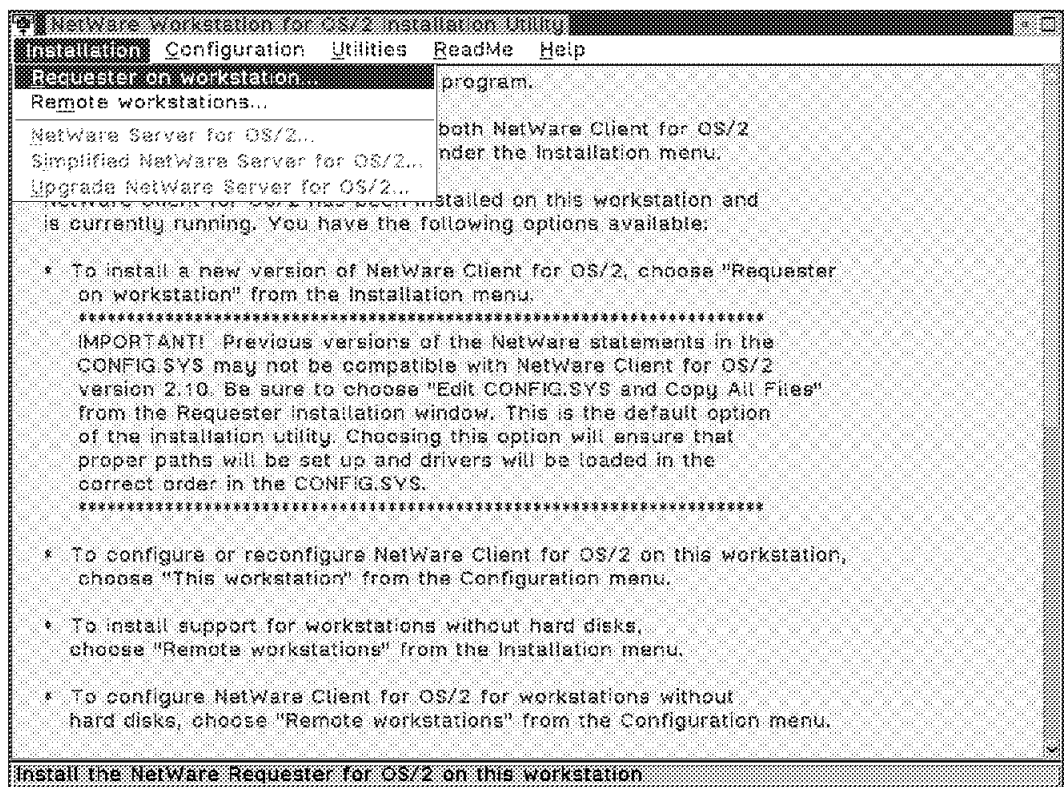


Figure 4. NetWare Installation Window

4. Set the target for the Requester files. You will require about 3.7MBytes for the files. We put the files in directory D:\Netware and our source drive was drive A. Click on the OK button, as shown in Figure 5 on page 13.

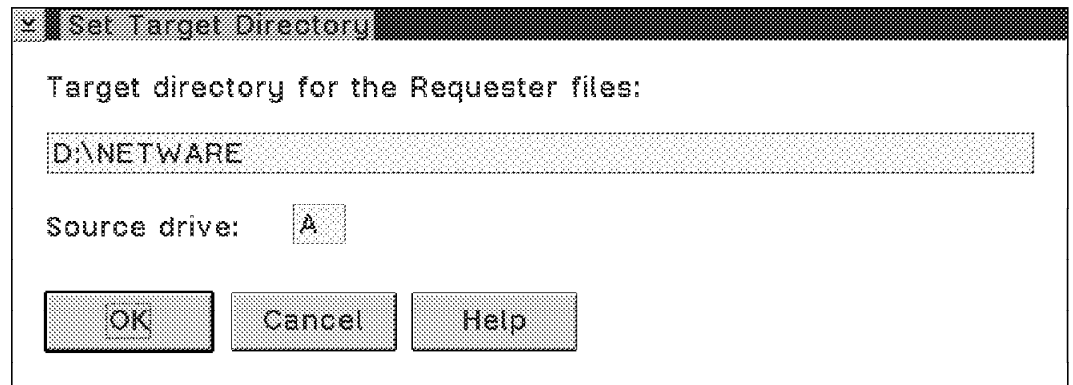


Figure 5. Selecting a Destination Drive for NetWare Files

5. Since we were doing a completely new installation we copied all the files and changed the CONFIG.SYS to include NetWare. Click on the **Edit CONFIG.SYS and Copy All Files...** radio button and click on the OK button as shown in Figure 6.

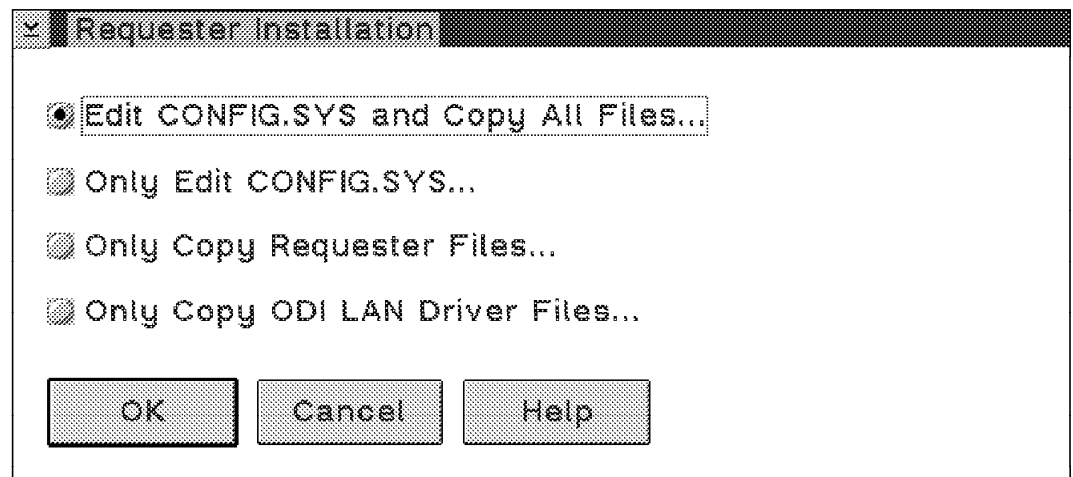


Figure 6. Type of Installation

6. Choose an ODI LAN driver. It does not matter which one you choose because with multiple protocols LAPS will place a REM in front of the driver in the CONFIG.SYS file and insert DEVICE=C:\IBMCOM\PROTOCOL\ODI2NDI.OS2 in its place. We chose the TOKEN.SYS driver as shown in Figure 7.

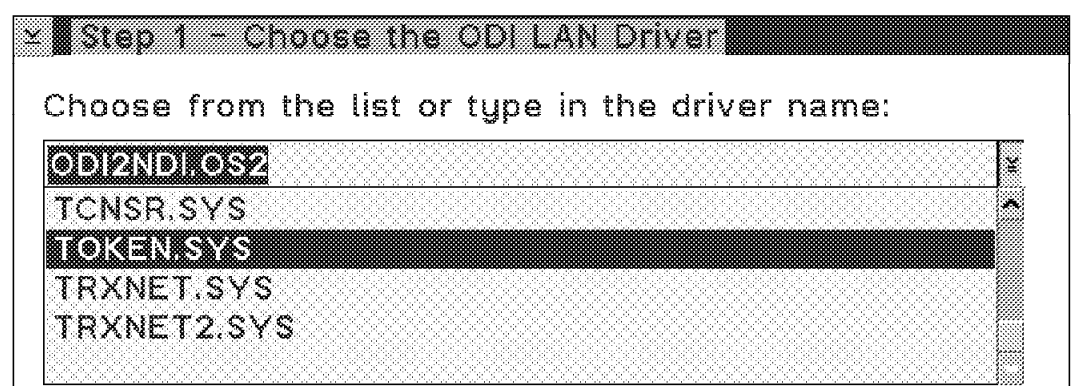


Figure 7. Selecting an ODI Driver

The resultant protocol stacks will be modified by LAPS to look similar to those shown in Figure 8 on page 14. The figure shows both token-ring and Ethernet.

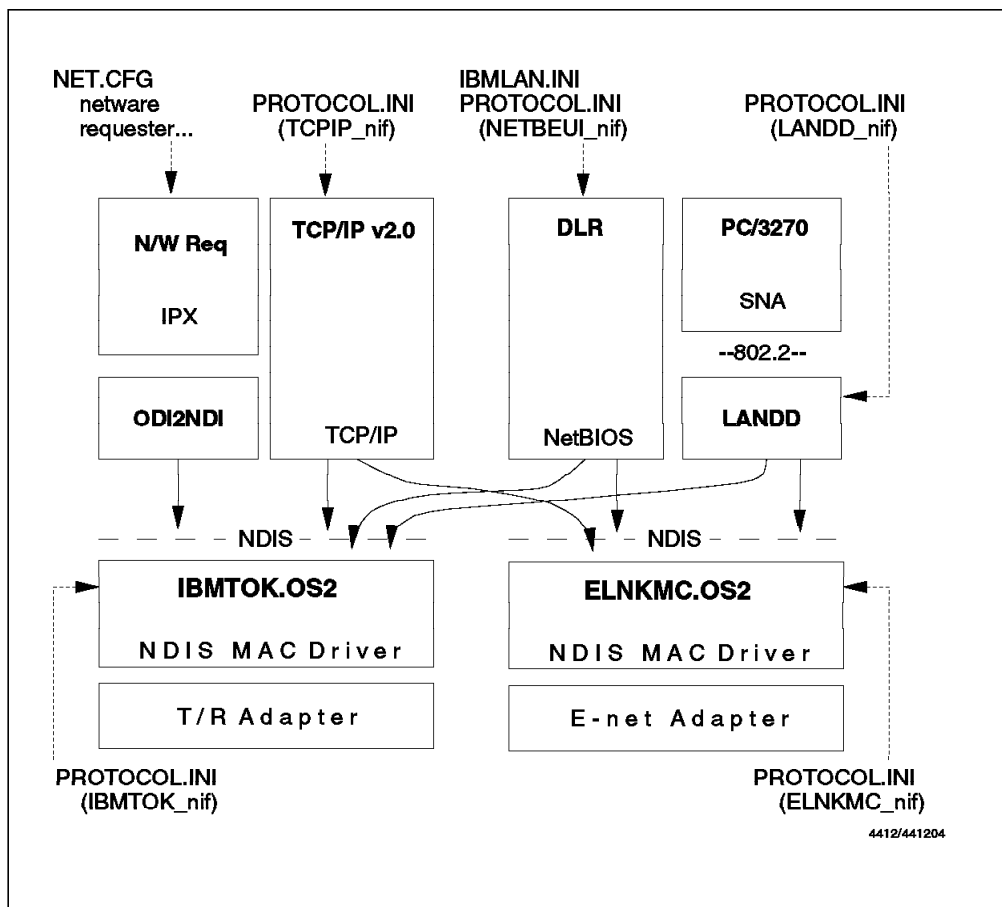


Figure 8. Protocol Stacks

The CONFIG.SYS NetWare section contained the following before we updated LAPS:

```
REM --- NetWare Requester statements BEGIN ---
.
.
.
DEVICE=C:\NETWARE\TOKEN.SYS
DEVICE=C:\NETWARE\ROUTE.SYS
.
.
.
REM --- NetWare Requester statements END ---
```

Figure 9. CONFIG.SYS Statements Added by NetWare

After updating it, it looked like this:

```

REM --- NetWare Requester statements BEGIN ---
.
.
.
DEVICE=C:\IBMCOM\PROTOCOL\ODI2NDI.OS2
REM -- ODI-Driver Files BEGIN --
REM DEVICE=C:\NETWARE\TOKEN.SYS
REM -- ODI-Driver Files END --
DEVICE=C:\NETWARE\ROUTE.SYS
.
.
.
REM --- NetWare Requester statements END ---

```

Figure 10. CONFIG.SYS after LAPS Has Been Run

After LAPS is run the NetWare installation is finished.

7. Click on the **continue** button.

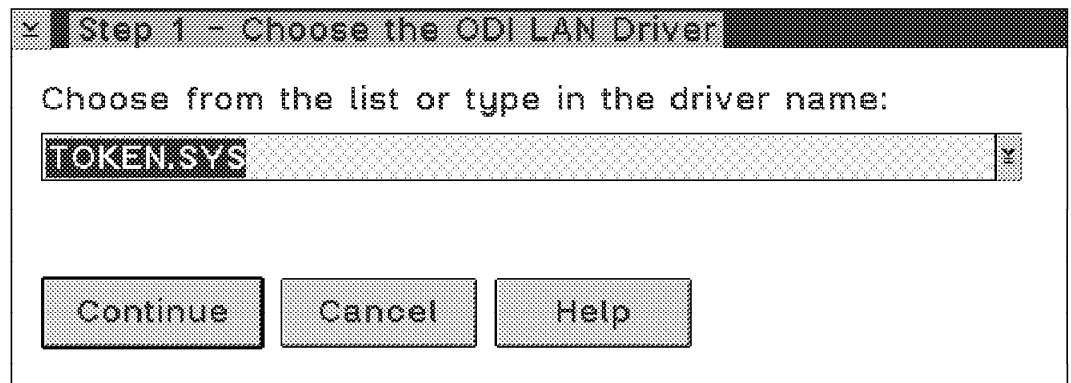


Figure 11. Selecting an ODI Driver 2

8. We left IPX Support for DOS and Windows Off and clicked on Continue, because it is not required in our setup. See Figure 12.



Figure 12. IPX Support for DOS and Windows

9. SPX support and Named Pipes Support are required. The default Client Support is sufficient. See Figure 13 on page 16.

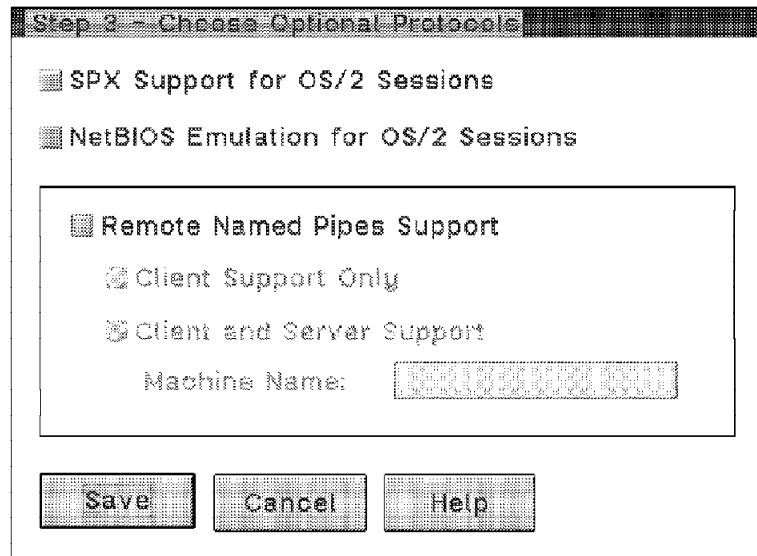


Figure 13. Optional Protocols

10. Click on Save.
11. Click on OK to save changes to the CONFIG.SYS file. See Figure 14.

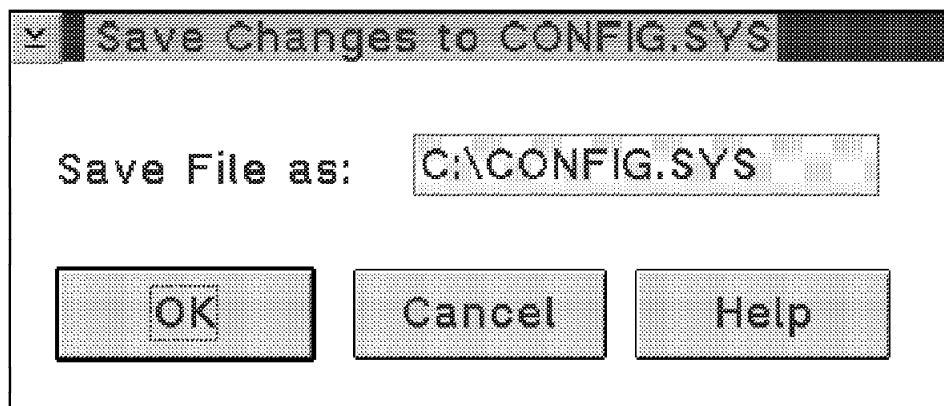


Figure 14. Save Changes to the CONFIG.SYS

12. Click on the **Copy all the drivers on the disk** radio button. It is better to copy all the drivers since important drivers like ROUTE.SYS are not copied by default. See Figure 15 on page 17.

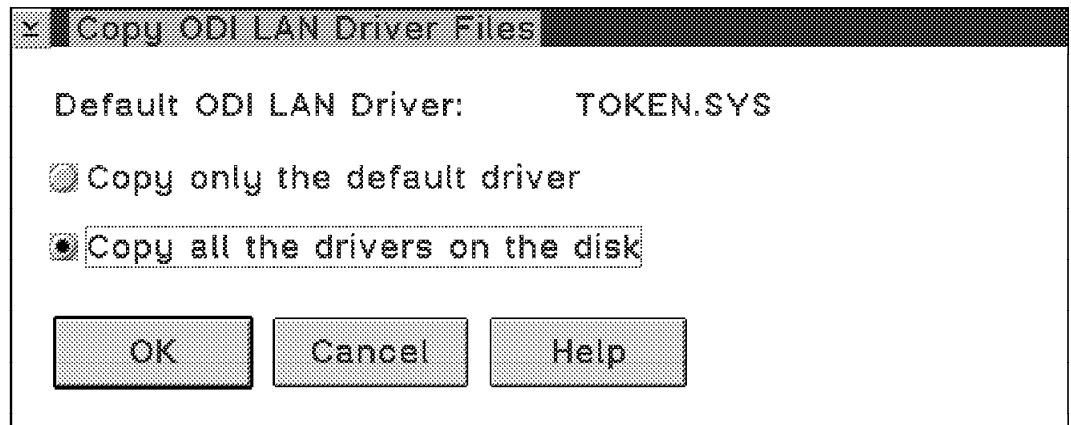


Figure 15. Copying All the Drivers on the Disk

13. A confirmation window as shown in Figure 16 will be displayed. To proceed with the installation click on the **Copy** button.



Figure 16. Copy Requester Files

14. A window with the location of the NET.CFG file will be displayed. This file is used to alter the default startup parameter of the requester. It should be located on the boot drive in the root directory. See Figure 17.

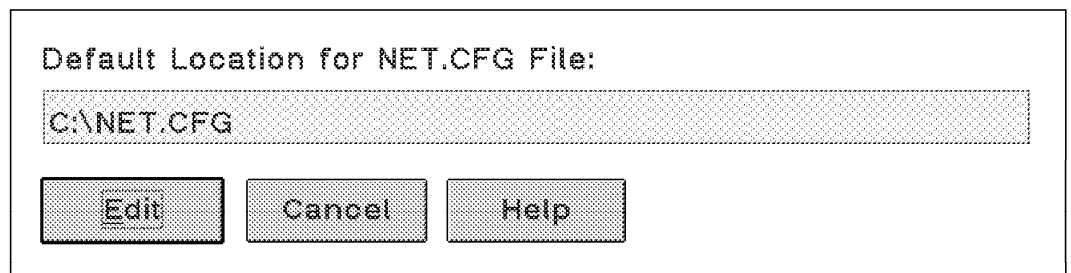


Figure 17. Location of NET.CFG File

15. A window for editing the NET.CFG file will appear. You should edit this file if:
- Your system has more than one network adapter.
 - You are using another frame type besides 802.2.
 - You want to change the packet security level.
 - You want to turn off packet burst or LIP transmissions.

- You are connecting to a token-ring using source routing.
- You want the workstation to connect to a preferred server.
- You are setting up remote booting workstations.

We used the default settings so we made no changes to the NET.CFG file and clicked on the **SAVE** button. See Figure 18.

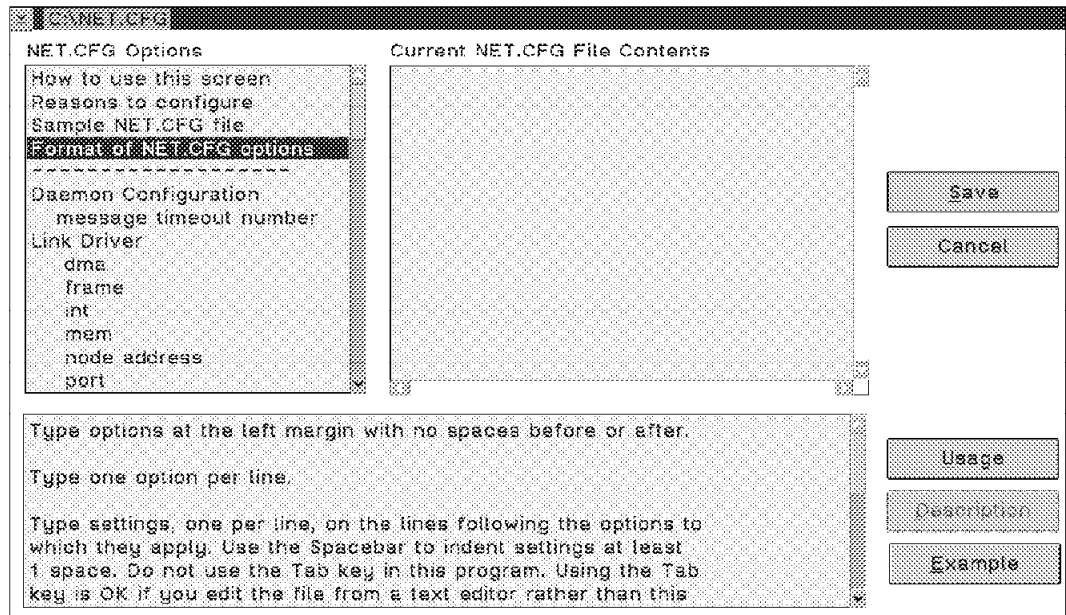


Figure 18. Editing the NET.CFG

Note that in LAPS you can make changes to:

- The network adapter address
- Frame header support
- ODI2NDI trace level

16. A window informing you that you have finished the installation will appear. See Figure 19 on page 19.

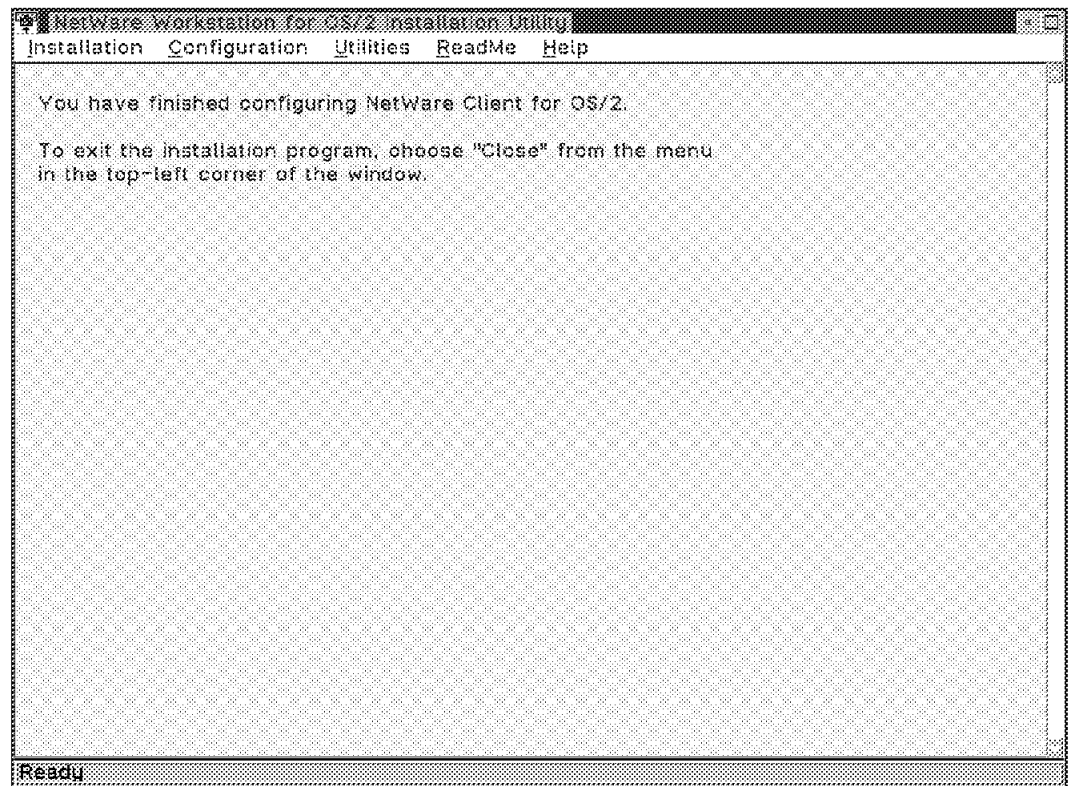


Figure 19. NetWare Requester Installed

17. To complete the installation and make changes to the CONFIG.SYS file run LAPS to add NetWare Requester Support. To see the final CONFIG.SYS file see Appendix C, "Configuration Files for Our NV2MGR1 Machine" on page 293.

2.5 IBM TCP/IP for OS/2 and Database Manager 2/2 Installation and Configuration

Following are the TCP/IP and Database Manager 2/2 configurations that were used. OS/2 TCP/IP and Database Manager 2/2 are required on the same machine as the LMU and LNM proxy agents. Following is the installation and configuration steps used in our environment.

2.5.1 Installing TCP/IP Protocol in LAPS

In addition to installing the TCP/IP product, LAPS requires TCP/IP drivers. This is done by installing the LAN Adapter and Protocol Support (LAPS) diskette provided with IBM TCP/IP for OS/2. If you already have installed LAPS and have TCP/IP protocols, you can skip the LAPS installation and continue with the configuration. The following screens will step you through the configuration of the TCP/IP drivers using LAPS.

1. To start the installation of LAPS enter A:LAPS from an OS/2 window with the LAPS diskette in drive A:. The screen shown in Figure 20 will be displayed. Click on the **install** button to start the installation which can be used to add the TCP/IP protocol in LAPS.

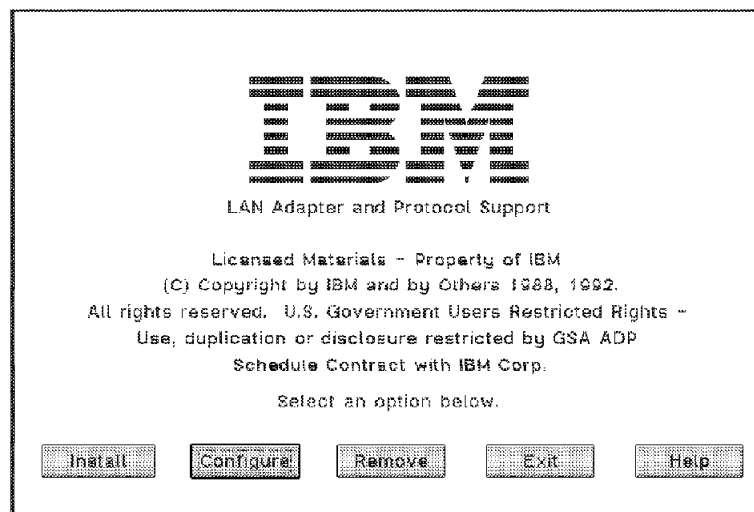


Figure 20. LAPS Installation Initial Screen

2. Click on **Install** to place the product on the C: drive. The progress of the installation will be shown.
3. Once LAPS is installed the initial screen will be displayed again as shown in Figure 20. Click on the **Configure** button to then configure and install the adapter and driver information that you need.
4. After clicking on the configure button, you will see the screen shown in Figure 21 on page 21. Click on the **Continue** button to configure LAPS.

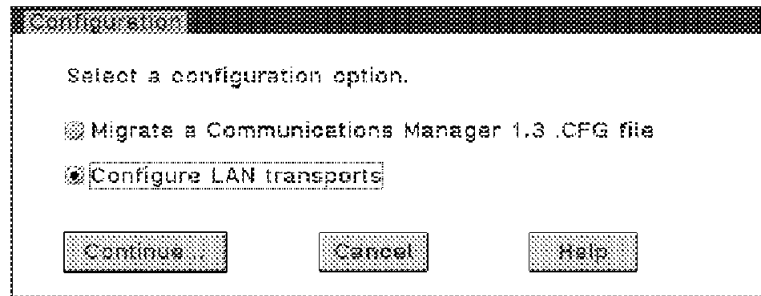


Figure 21. LAPS Configuration Option

Choose the appropriate information from the lists boxes as shown in Figure 22.

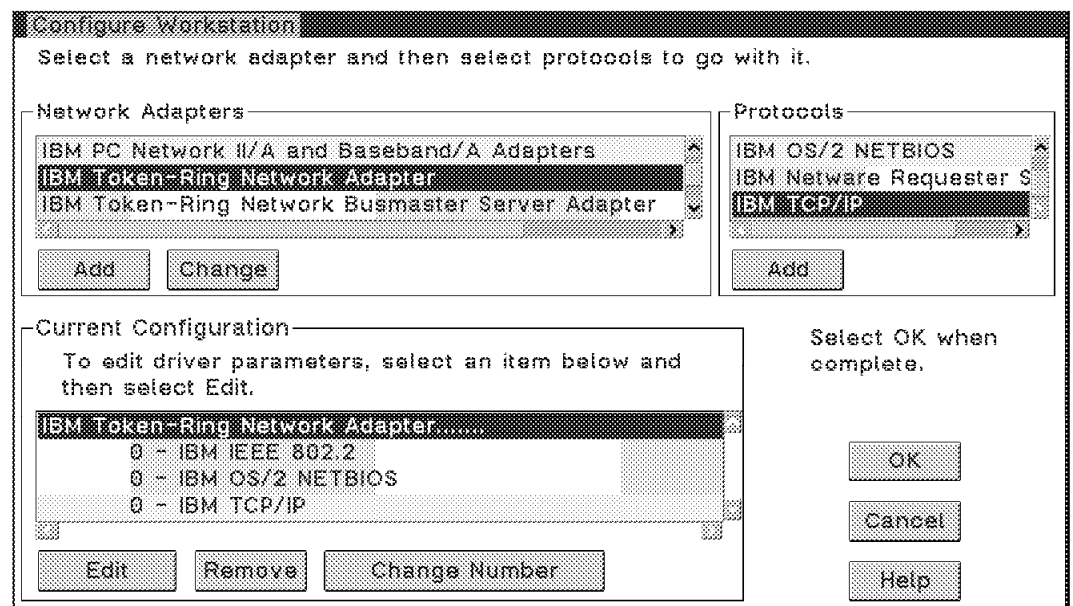


Figure 22. LAPS Main Screen

The following LAN configuration information is required:

1. **Network Adapter:** Choose the LAN adapter installed in your OS/2 workstation by selecting the adapter from the list and clicking on the **Add** button.
2. **Protocols:** Choose the TCP/IP protocol and then click on the **Add** button to install the TCP/IP drivers and associate it with the LAN adapter. In our case we installed all protocols that were available.

Note

If you do not have the TCP/IP option in the *Protocols* box, you will have to reinstall LAPS with the diskette supplied in your TCP/IP package to add this support. Go to step one with this diskette.

The Current Configuration list box shows what has been selected. We have shown an IBM token-ring adapter with TCP/IP, NetBIOS, and NetWare protocols installed. You need a minimum of one LAN adapter, but there will be times that

you will use more than one. Remember to edit all the protocols and the adapter to include a locally administered address.

Note

If you are using a token-ring adapter precede the network addresses with a capital "T" as shown in Figure 23.

Parameters for IBM IEEE 802.2	
Edit the parameters as needed. Except for parameters preceded by "*", changes affect all instances of the driver.	
*Network adapter address	T400052005103
*Type of Ethernet driver support	D
*System key value	0
*Adapter open options	2000
*802 trace level	0
*Maximum link stations	64
*Maximum SAPs	40
*Maximum group SAPs	0
*Maximum number of users	10
*Group 1 inactivity timer - Ti	255
*Group 1 response timer - T1	15
*Group 1 acknowledgment timer - T2	3
*Group 2 inactivity timer - Ti	255
*Group 2 response timer - T1	25

OK Cancel Range Help

Figure 23. Token-Ring Address

Select the **OK** button to update the LAPS configuration files with the new information. The files used by LAPS to store the information are:

C:\CONFIG.SYS
C:\IBMCOM\PROTOCOL.INI

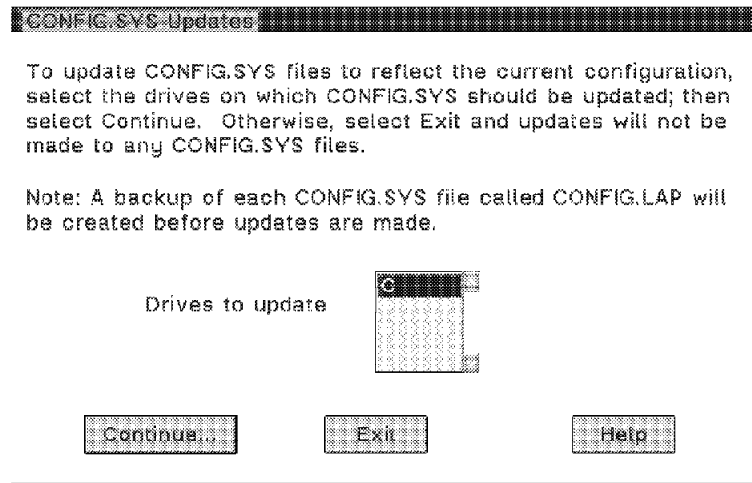


Figure 24. LAPS Start Update

Click on **Continue...** in the CONFIG.SYS panel as shown in Figure 24 to make the changes to these files.

Note: For a complete listing of the PROTOCOL.INI and the CONFIG.SYS of our NetView for OS/2 Managing Station (NV2MGR1), please see Appendix C, "Configuration Files for Our NV2MGR1 Machine" on page 293.

2.5.2 Installing TCP/IP

Installation of TCP/IP is a two-stage process. First, you install the base code for TCP/IP and apply the CSDs. Secondly, configure TCP/IP from the TCP/IP configuration notebook in the TCP/IP folder on the desktop.

The installation and configuration steps required are:

1. Install the TCP/IP drivers for the LAN adapters using LAPS, see 2.5.1, "Installing TCP/IP Protocol in LAPS" on page 20 Installing TCP/IP protocol in LAPS.
2. Install IBM TCP/IP for OS/2 and apply the CSDs
3. Configure the IBM TCP/IP for OS/2 information from the desktop notebook

The information you need to change is:

- a. IP address
 - b. Default router information
 - c. SNMP information - system contact and location, community name and trap destination
 - d. Host name
 - e. If you are going to issue remote commands using the LMU agent, you will also need to set up TCP/IP
1. Install TCP/IP on your machine by typing A:\TCPINST in an OS/2 window with disk 1 of the Base Kit in drive A:. On the first screen (see Figure 25 on page 24) select the options you wish to install. Select the Base Kit and all

others that you wish to install. We installed PMX (the XWindows client for AIX) systems because we wanted to do some work with the RISC System/6000. When LAPS is finished, you will be prompted to insert diskettes until the installation is finished. Use the Install/Run LAN Adapter and Protocol Support box if you have not installed LAPS. You will be prompted to install LAN Adapter Protocol Support (LAPS). Go to Installing LAPS on page 20. Once the TCP/IP product is installed the latest CSDs will need to be applied. The CSD numbers for all of OS/2 TCP/IP V2.0 are:

TCP20CSD - Base CSD	UN64092
DOS20CSD - DOS Access CSD	UN57546
NFS20CSD - Network File System CSD	UN57064
APP20CSD - Applications Only CSD	*Same as Base CSD Now*
PMX20CSD - X Server CSD	UN60006
XCL20CSD - X Client CSD	UN59347
XNT20CSD - Extended Networking CSD	UN60005
DNS20CSD - Domain Name Server CSD	UN60004
PGM20CSD - Programmers Tool Kit CSD	UN57887
NET20CSD - RFC NetBIOS REL 1.0 CSD	UB09131

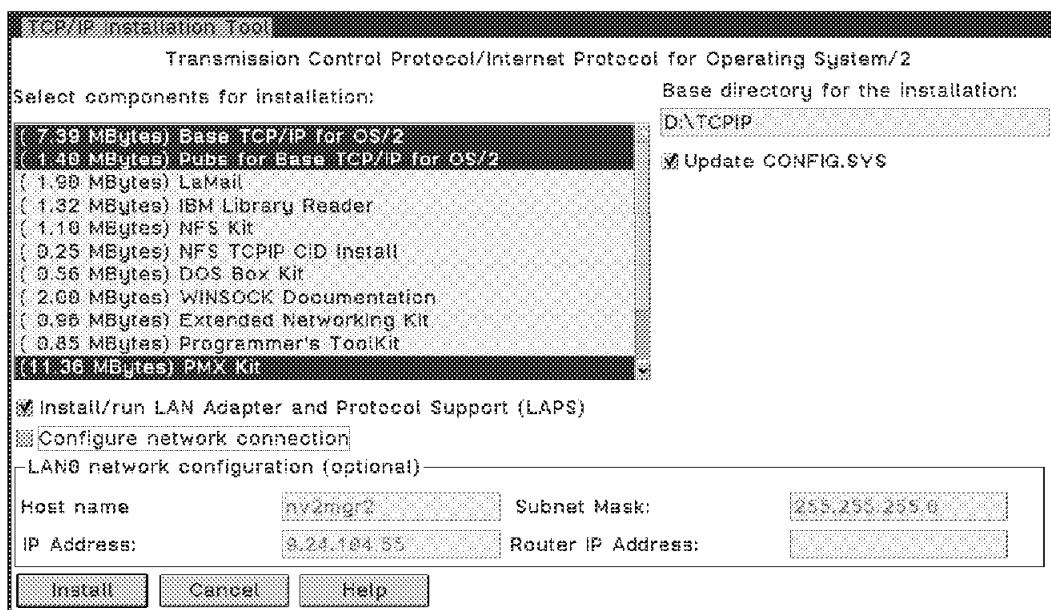


Figure 25. Installation Screen for TCP/IP

Note: You may not have all of the packages listed in the scroll box of Figure 25. They are not necessary for use with NetView for OS/2 but can be purchased separately.

2. From the desktop open the TCP/IP folder.

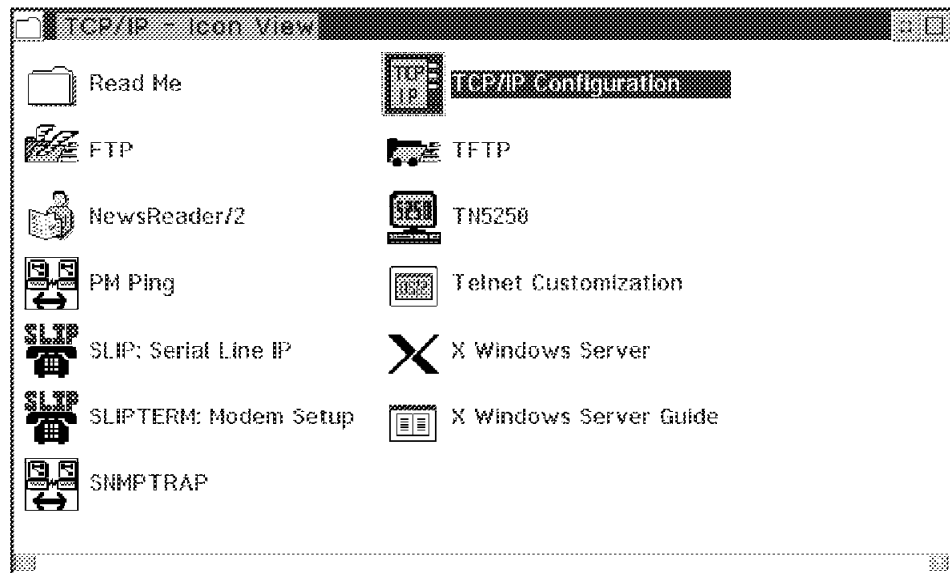


Figure 26. Icon View of TCP/IP Features

3. Double click on the **TCP/IP Configuration** icon in the TCP/IP folder as shown in Figure 26. The first panel will be the Configure Network Interface Parameters panel. It is page 1 of 8 and each page represents one network adapter. We had only one adapter so only the first page had to be customized.

2.5.2.1 Network

The following steps must be done to configure the network pages of the configuration notebook. See Figure 27.

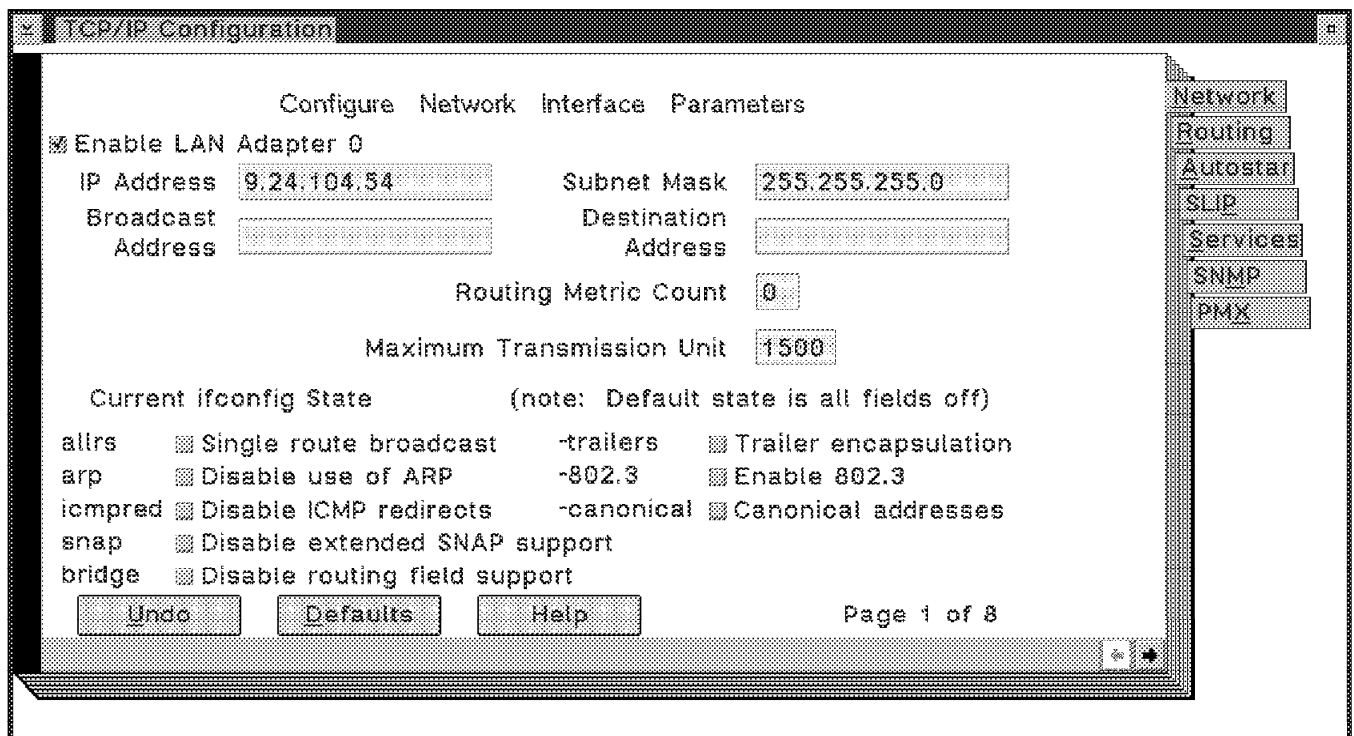


Figure 27. TPC/IP Configuration Notebook: Network

1. Place a check mark in the Enable LAN Adapter 0 box by clicking on it.
2. Enter the IP address of the workstation on which TCP/IP is being installed.
The IP addresses of our two OS/2 workstations were:
 - NV2MGR1 was defined as 9.24.104.54 with a subnet mask of 255.255.255.0
 - NV2MGR2 was defined as 9.24.104.55 with a subnet mask of 255.255.255.0
3. The subnet mask is defined by the network coordinator for your network.
4. The routing metric count is the number of jumps that can be used to access another IP address. We left it at 0, the default value.
5. If you have another adapter installed on your machine you should fill in the second page (2 of 8).

2.5.2.2 Router Name Server Information

The following screens show how to configure routers and domain name servers in your network if you are using them.

To configure the router take the following steps:

1. Click on the Routing tab to bring up the Routing information window. See Figure 28.

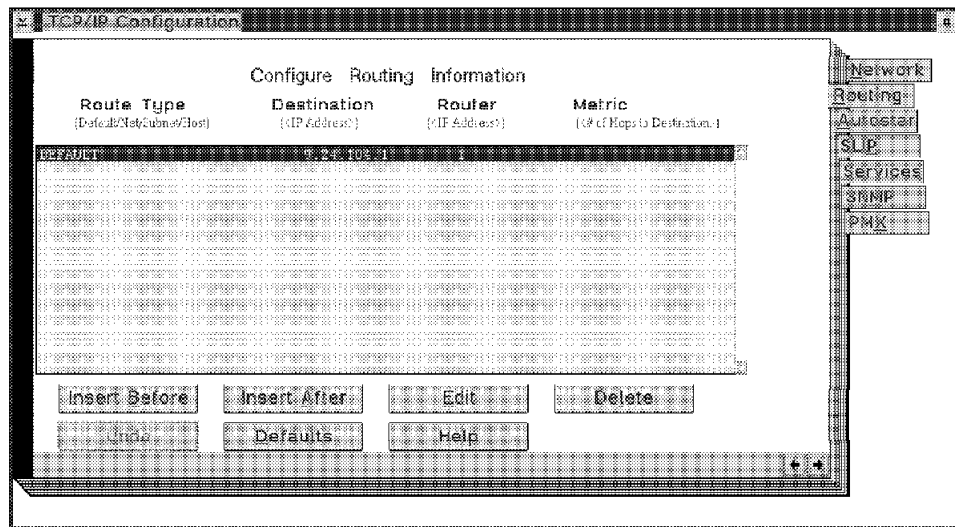


Figure 28. TCP/IP Routing Configuration

2. Highlight the default definition and click on the **Insert Before** button. A Route Entry - Edit window will be displayed as shown in Figure 29 on page 27.

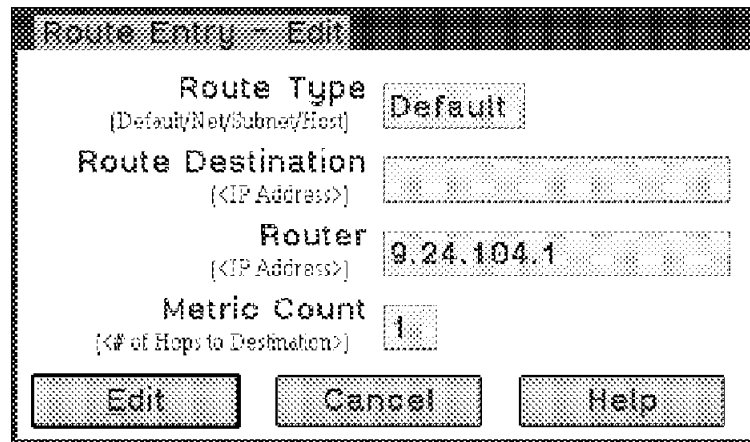


Figure 29. Entry Window for Router Information

3. Specify a router type. Valid values for Router Type are Net, Subnet, Host or Default.
4. The Route Destination is the IP address you would like to access. This can be a subnet address for subnet or a host address. We left it blank.
5. We defined one default router used by our network. The IP address of our default router was 9.24.104.1

2.5.2.3 Domain Name Services

Click on the **Services** tab of the configuration notebook to configure the domain name server. See Figure 30.

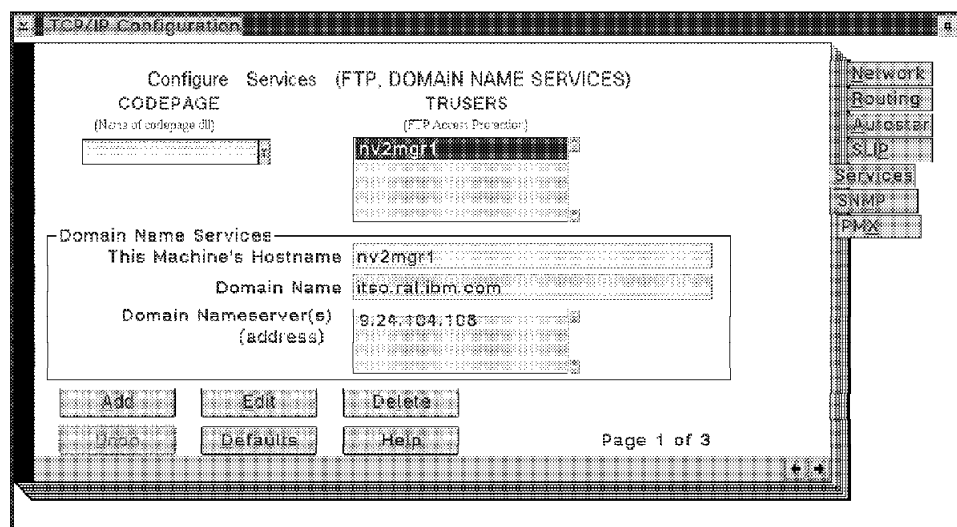


Figure 30. TCP/IP Hostname and Other Services Configuration

1. Enter the host machine name in the **This Machine's Hostname** field. This is the name given to one of our OS/2 workstations. In our environment the name was:

- NV2MGR1

This updates the C:\CONFIG.SYS file on the NV2MGR1 machine with the following entry:

SET HOSTNAME=NV2MGR1

2. Enter the domain name in the **Doman Nameserver** field. Our domain name is itso.ral.ibm.com.
3. Enter the domain server's TCP/IP address in the **Domain Nameservers (address)** window. We used 9.24.104.108.

Note: You may have to contact your on-site TCP/IP LAN administrator to find out the name and address of your domain name server.

FTP access

To enable other managing machines to copy your seed file to their machine using TCP/IP FTP, do the following in the configuration notebook:

1. In **Services** on page 1 enter the host name of the machine that will be allowed access.
2. In **Services** on page 3 enter a Telnet password in the password field. The password will not be displayed, but it will show up in your CONFIG.SYS file.
3. In **Autostart** on page 1:
 - a. Enable this machine to start the inetd super server.
 - b. Enable other users to log in to this machine.
 - c. Enable others to access your files by using FTP.

You should now be able to access this machine remotely.

2.5.2.4 SNMP Information

The management information flow between NetView for OS/2 and OS/2 workstations is done using the SNMP protocol.

The SNMP information required is:

1. System contact and location (page 1 of 2 on the SNMP tab)

The information entered for these MIB-II variables, SYSCONTACT and SYSLOCATION, for each computer as shown in Figure 31 on page 29, are:

- a. Mirek Iwachow and Raleigh, bld 062, room L610

This updates the C:\CONFIG.SYS file with the following two entries:

SET SYSCONTACT=Mirek Iwachow

SET SYSLOCATION=Raleigh, bld 062, room L610

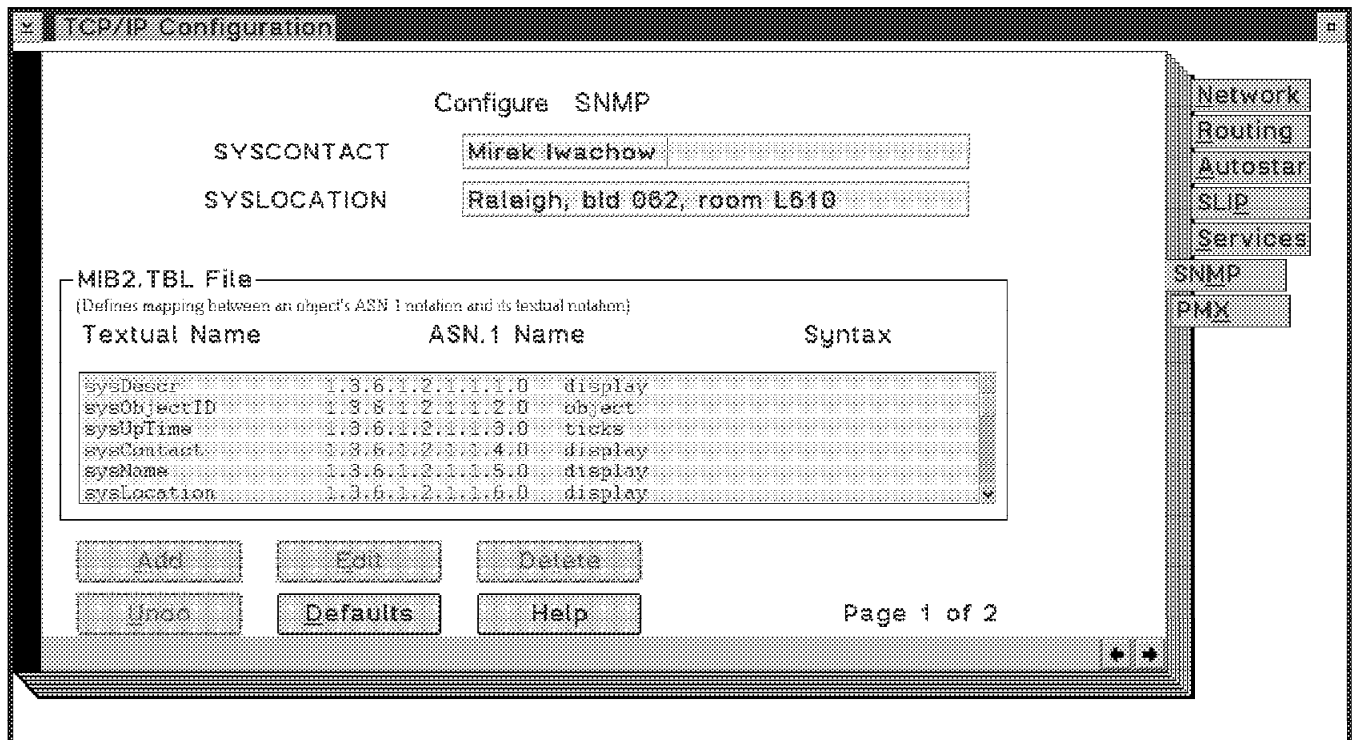


Figure 31. TCP/IP SNMP MIB-II Information

Click on the arrows to select the second page of the SNMP configuration tab as shown in Figure 32 on page 30.

2. SNMP trap destination (page 2 of 2 on the SNMP tab)

The IP address of the managing NetView for OS/2 is needed to allow SNMP traps generated by the OS/2 workstations to be sent to the manager, as shown in Figure 32 on page 30. The host name of the manager can be entered if the D:\TCPIP\ETC\HOSTS file (or Domain Nameservers) have the mapping of the name (in our environment NV2MGR1) to the actual IP address. You can have multiple trap destinations. You might have multiple trap destinations if you are setting up your management environment to include a backup manager.

The file D:\TCPIP\ETC\SNMPTRAP.DST is updated with this information:

9.24.104.55 UDP

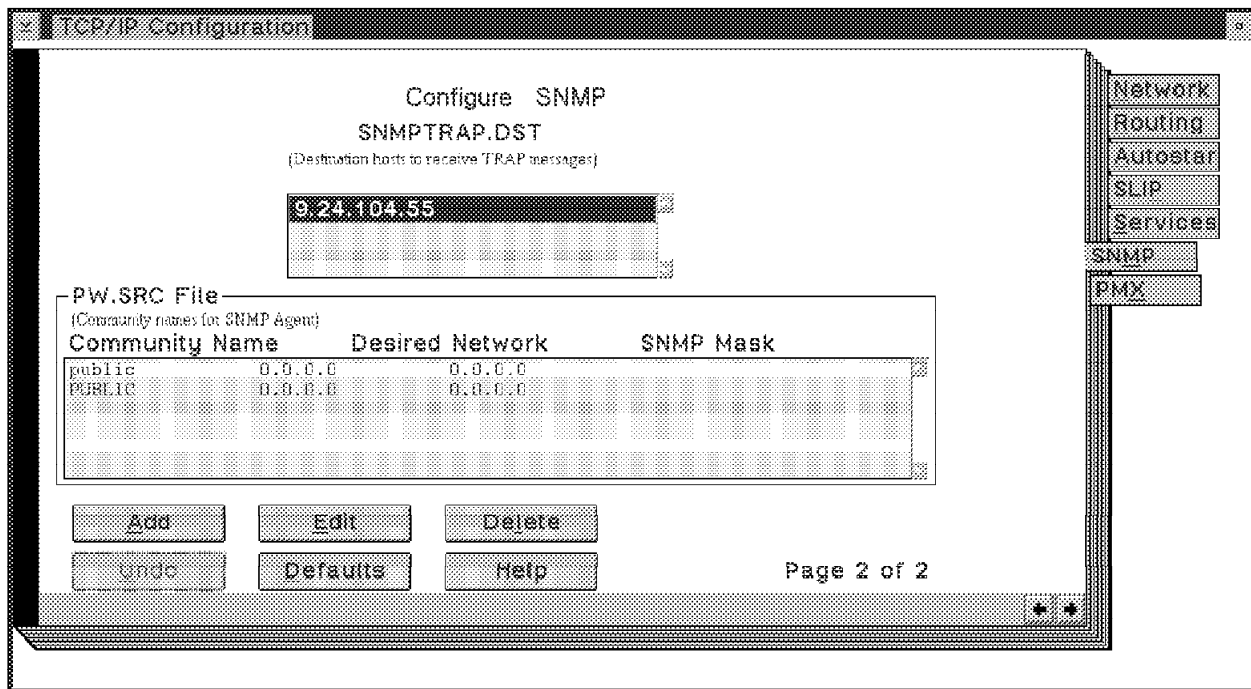


Figure 32. TCP/IP SNMP Traps and PW.SRC File

3. SNMP community name

The community name is required and must match the community information on the manager. This provides the authority for the manager to access the MIB information on the OS/2 workstations and receive the SNMP traps generated as shown in Figure 32. This is how SNMP V1 provides security.

The following file D:\TCPIP\ETC\PW.SRC is updated with this information:

```
public      0.0.0.0      0.0.0.0
PUBLIC      0.0.0.0      0.0.0.0
```

The command `MAKE_PW` needs to be run against the PW.SRC file to compile the information. You need to make the ETC directory your current directory and then execute the command as shown:

- a. `CD \TCPIP\ETC`
- b. `MAKE_PW`

The following message will be returned if `MAKE_PW` compiled successfully:

I have written 2 entries into "snmp.pw"

2.5.2.5 Starting TCP/IP

To ensure that TCP/IP is automatically started when OS/2 is started you can update the OS/2 startup file, C:\STARTUP.CMD, or drag the OS/2 TCP/IP Startup icon into the OS/2 Startup folder. Following is what is required to automatically start TCP/IP using the C:\STARTUP.CMD file:

```
REM * OS/2 TCP/IP startup command file
REM *****
call tcpstart.cmd
```

The option to start TCP/IP through the OS/2 Startup folder was not used, since the SNMP multiple protocol daemon requires that TCP/IP is started before it is loaded.

Since CONFIG.SYS is updated during the customization of OS/2 TCP/IP the OS/2 workstation will need to be restarted to complete the installation.

2.5.3 OS/2 Database Manager 2/2 Installation

IBM Database Manager 2/2 is required for the OS/2 workstations to store the information gathered from the network. The version of DB2/2 that we installed was Version 1.01. After applying the latest CSD we brought the code up to level WR07025.

The steps required to install it are:

1. Configure the type of installation - Client with local databases
2. Configure workstation name - NV2MGR1

To start the Database Manager 2/2 installation, type A:DBINST from an OS/2 window. You will see the *DATABASE 2 OS/2 Install* window as shown in Figure 33.

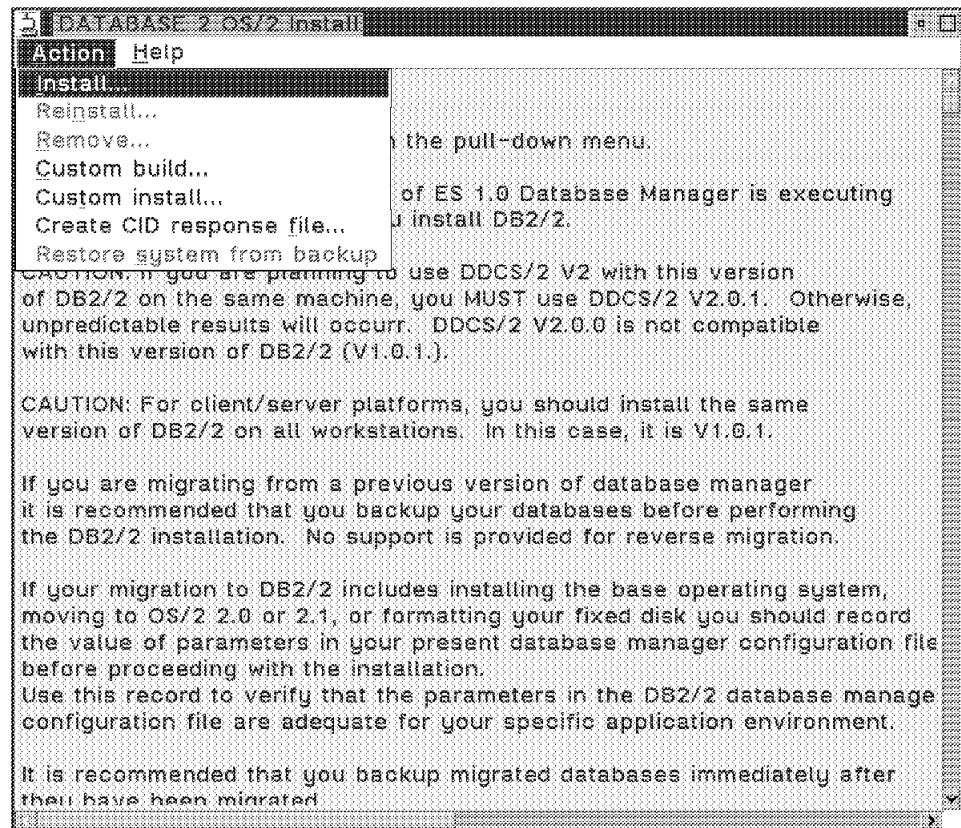


Figure 33. Database Manager 2/2 Installation Initial Screen

To start the installation process, choose **Action..Install** from the pull-down menu. The screen shown in Figure 34 on page 32 is displayed with the following options already selected:

- Target Drive: Choose the drive location for the Database Manager 2/2 code. In our environment, drive **D** was selected. It is important to note that you must have approximately 3MB of free disk space on drive C: for temporary files.

- Type: Select **Client with local database**. The *Client* option means you have no local Database but you can access one over the network. The *Standalone* option means you can only use the database on this machine. The *Client with local database* option means the database resides locally and other users can access it. Since LMU wants to share the database with clients, this is the required selection.
- Features: The other options such as *Query Manager*, *Database Tools* and *Documentation* were chosen. However, they are not required.

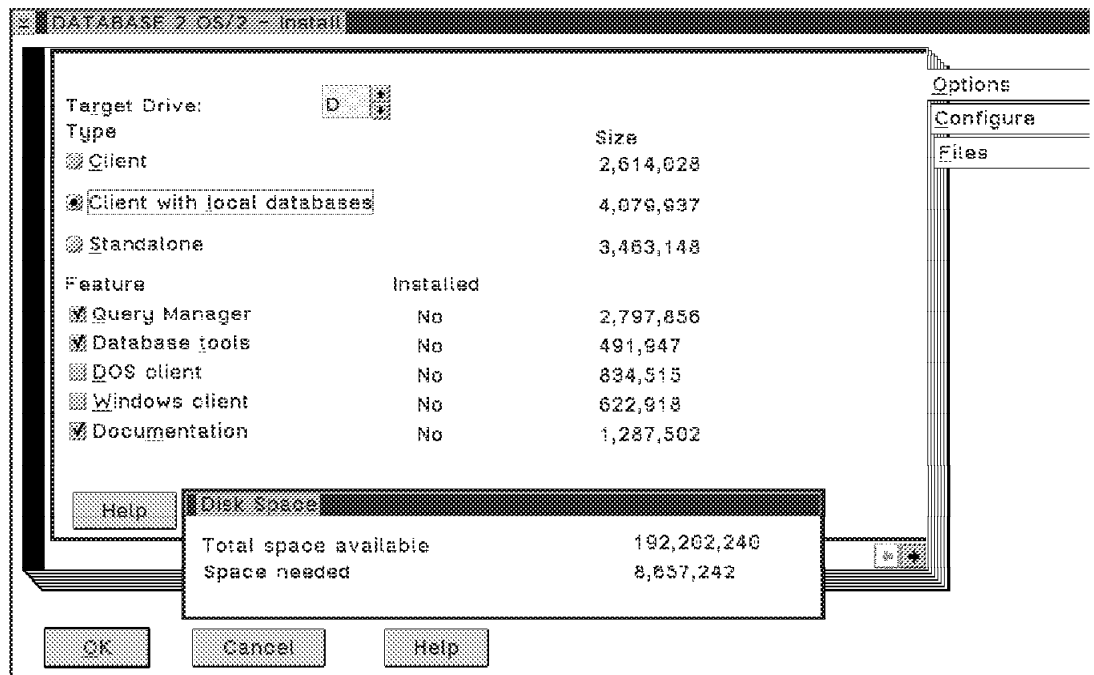


Figure 34. Database Manager 2/2 Options Installation

A workstation name is required. This name is not checked in any of the other configurations.

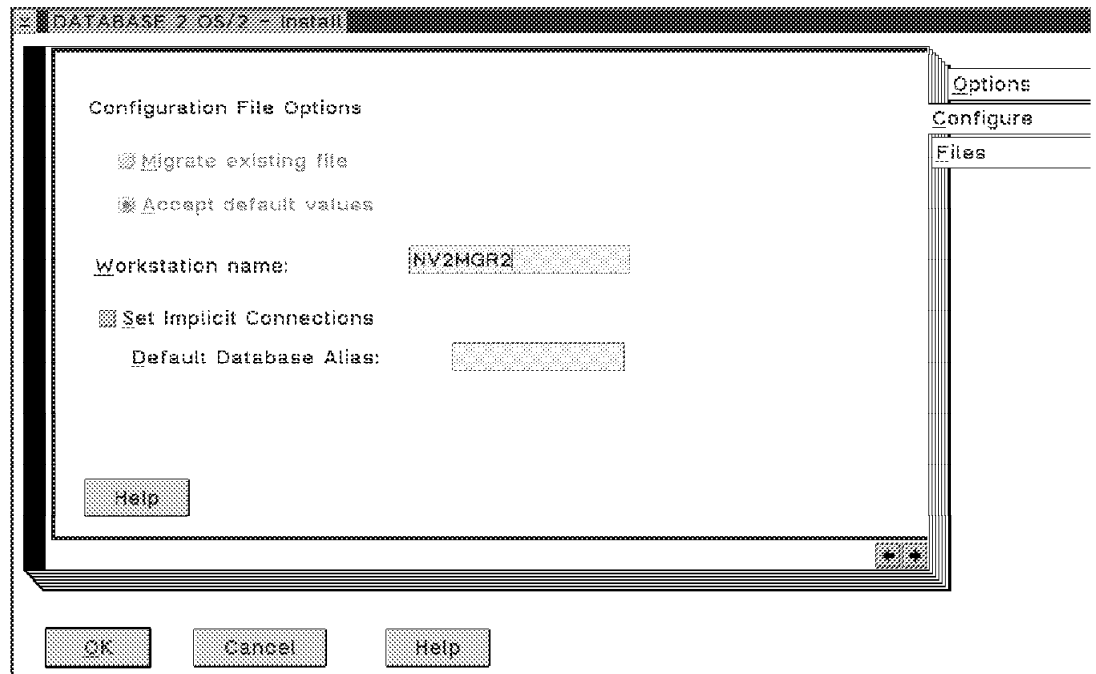


Figure 35. Database Manager 2/2 Configure Installation

Once you have entered the above information, select the **OK** button to start the installation. A screen is displayed showing the progress of the installation process. You will then be prompted for each additional diskette.

Once the installation has completed the message shown in Figure 36 is displayed.

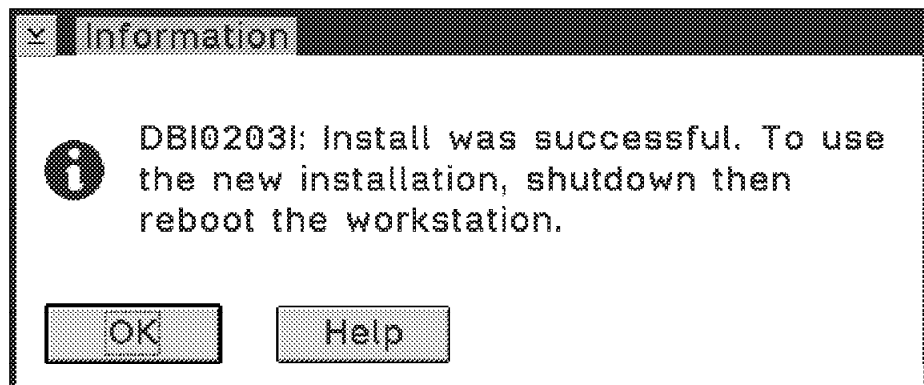


Figure 36. Database Manager 2/2 Installation Successful

The OS/2 workstation needs to be restarted to complete the installation of the Database Manager 2/2 product. The IBM Database Manager 2/2 is required to store information gathered from the network. The Database Manager in our case is installed on the managing station.

2.6 Installing and Configuring NetView for OS/2 Managed Systems

During our testing of the installation procedures, we found the *NetView for OS/2 Installation and Administration Guide, SC31-8100* to be very useful. This guide shows the administrator which panels will be displayed during installation, using a step-by-step approach. This document will not attempt to duplicate what was done in that document; however, it will give some hints and tips on what to look out for.

2.6.1 Configuring the NetView for OS/2 Agent on a Managed System

At the beginning of the installation process, the installation program will prompt you for what components of NetView for OS/2 you want to install. You are presented with a list of components as shown in Figure 37. To install and configure the Agent code, choose the line item entitled **Agent for OS/2**.

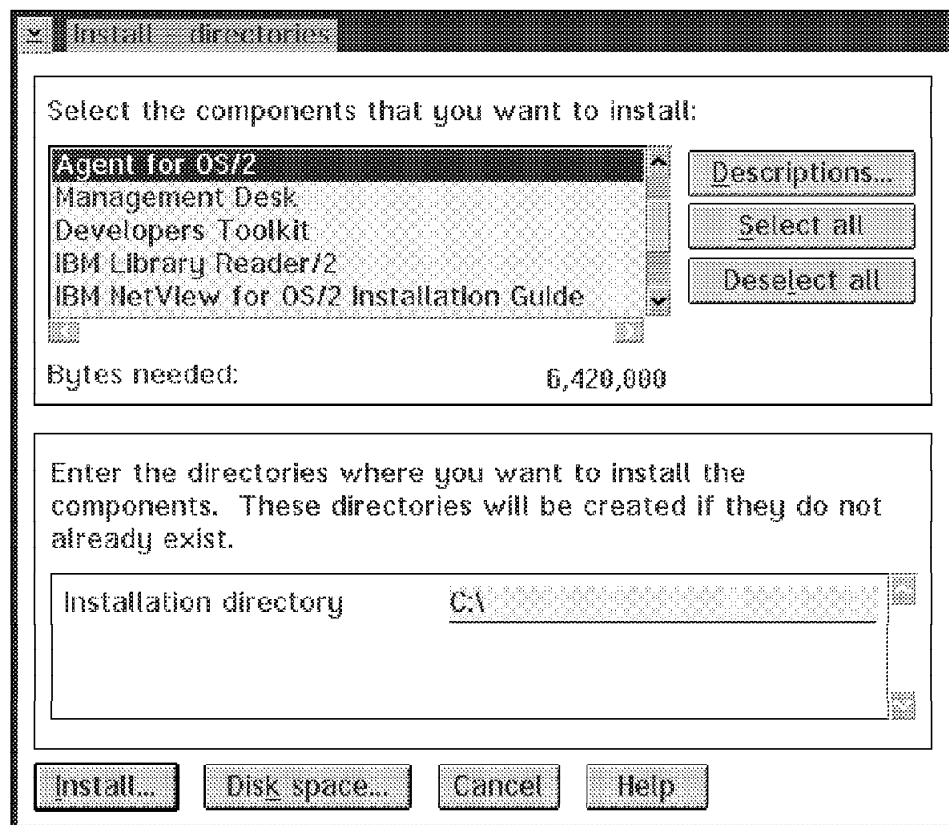


Figure 37. NetView for OS/2 Installation Directories - Agent Code

To install the documentation for the NetView for OS/2 Agents, select the line item entitled **IBM NetView for OS/2 Agents** as shown in Figure 38 on page 35. In fact, all the NetView for OS/2 online documentation is listed after the **IBM Library Reader/2** line item.

Also in this panel, do not forget to change your installation directory to some other logical drive if you do not want to install on the C:\ directory. The default root directory for NetView for OS/2 is C:\ANV2.

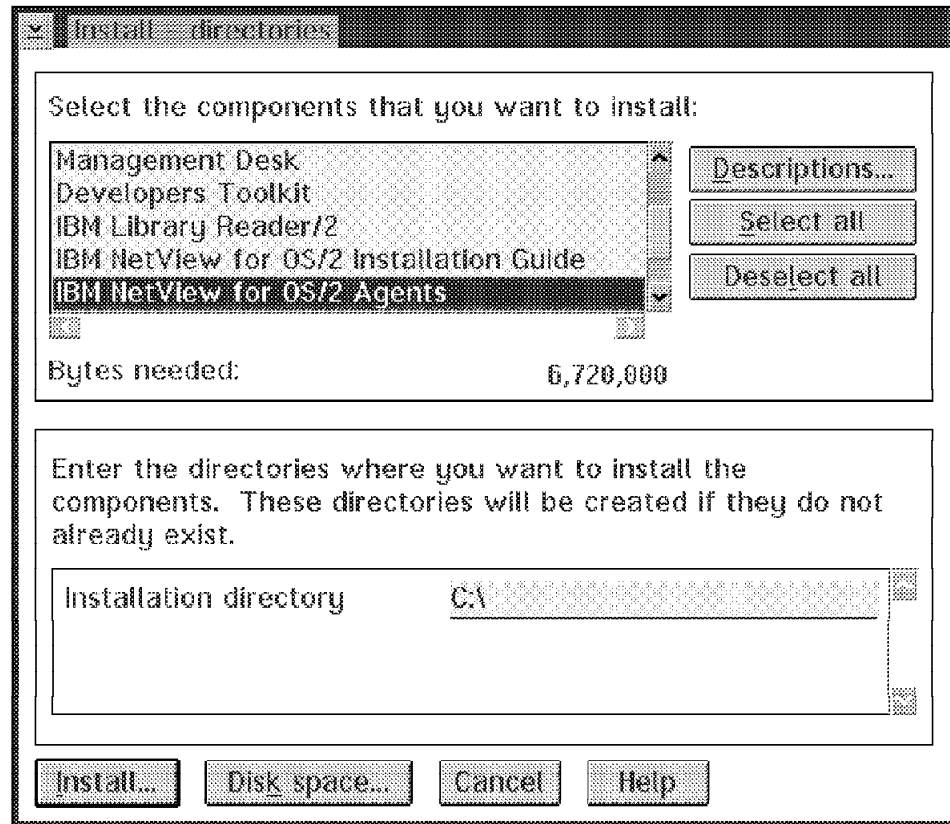


Figure 38. NetView for OS/2 Installation Directories - Agent Documentation

After you have selected all your components and changed to the appropriate Installation directory, click on the **Install...** button to start installation.

When presented with the Installation Transports panel as shown in Figure 39 on page 36, remember that even though you may have all three transports (IP (TCP/IP, AnyNet/2), IPX and NetBIOS) installed, you can only pick *one* transport to use to communicate to a managing station.

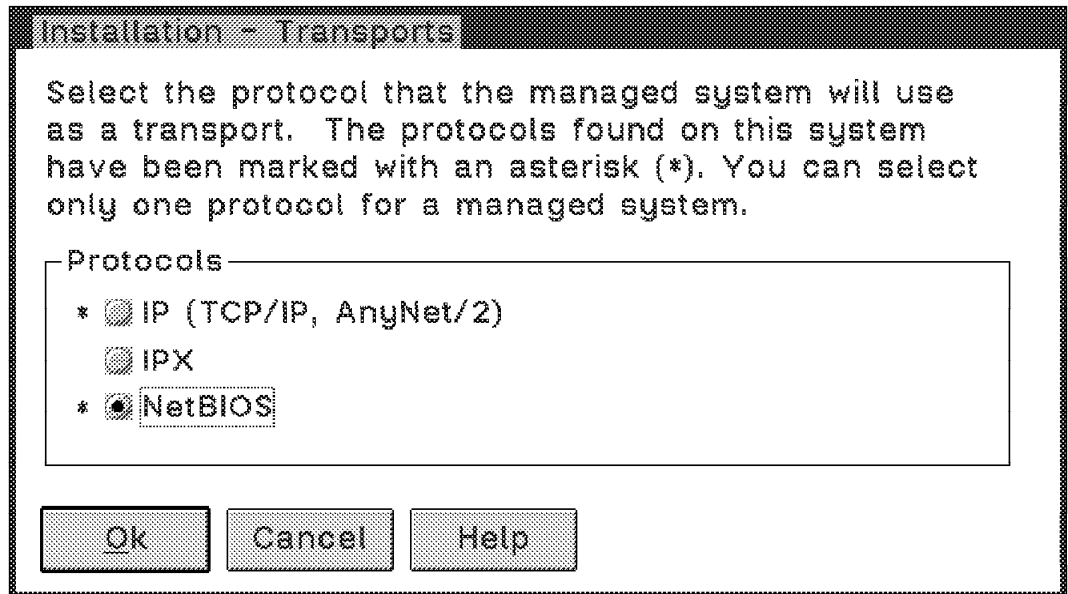


Figure 39. NetView for OS/2 Installation Transports for Agents

2.6.2 Implementing Security on the Managed System

You can use the facilities of SNMP for implementing security on the managed systems. In most cases, you would want to limit the number of managing stations that can have access to the managed station's MIB variables. With the NetView for OS/2 SNMP Configuration panels that you see during installation, you can specify who your managing station is, thus preventing any other system from viewing and updating the managed station's MIB entries.

There are two SNMP Configuration windows required for setting up an OS/2 Agent.

The first window that will come up will be to set up the community name, trap destinations and specify the protocols that will be used. When presented with the Community Names panel as shown in Figure 40 on page 37, the administrator can define what kind of access managing stations can have, and what managing stations can access this Agent station.

If you are certain that this Agent's MIB variables can be updated, then click on the **read/write** radio button. Also, if you wanted all managing stations in the **public** community to have access to this machine, then enter 0.0.0.0 in both the Address field and the Network Mask field.

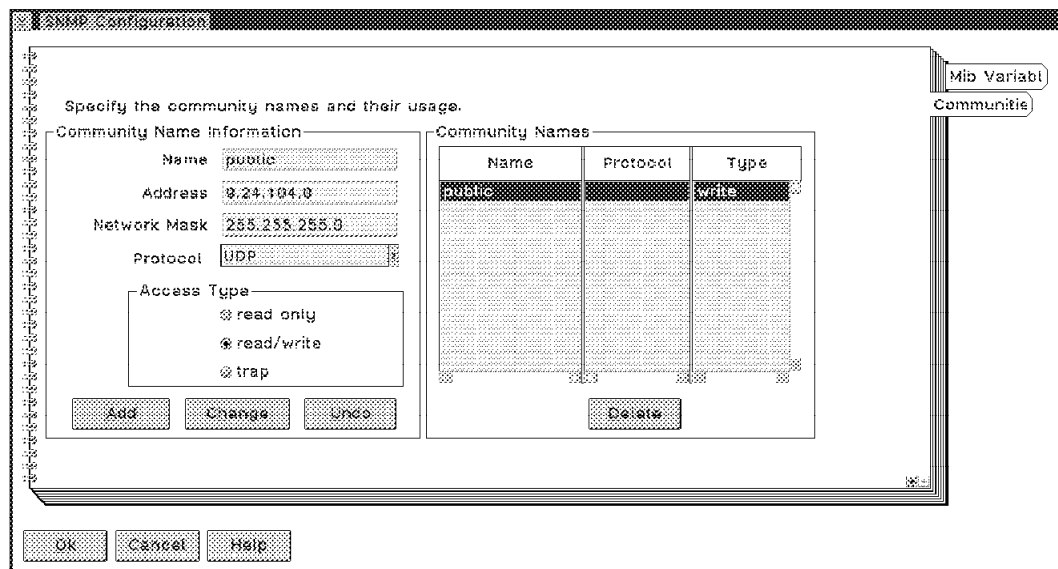


Figure 40. SNMP Community Names for Agents - TCP/IP Environment

If you want to limit the managing stations to those addresses that begin with 9.24.104, then you would put 9.24.104.0 in the Address field and 255.255.255.0 in the Network Mask field as shown in Figure 40. To add this security entry, click on the **Add** button and it will add a new line to the *Community Names* list box at the right side of the window.

If you want to limit the managing station to one particular address such as 9.24.104.54, then you would put 9.24.104.54 in the Address field and 255.255.255.255 in the Network Mask field. As you can see, if the result of a logical AND between the manager's address and the Network Mask is what is in the Address field, then NetView for OS/2 will allow the SNMP request to take place.

2.6.2.1 Implementing Security on a NetBIOS Managed System

If you were setting up this agent in a NetBIOS environment, you would click on the **Protocol** drop box, and select the **NetBIOS** line item. You can then select a Community Name. In our case, we entered the name **public** to be consistent with our TCP/IP machines. Again, if you want to limit the number of managing stations with access to this resource, just ensure that the managed and managing stations all reside in the same community.

For update authority, select the **read/write** radio button, and then click on the **Add** button. Notice that a new line item is added to the *Community Names* list box at the right side of the window as shown in Figure 41 on page 38.

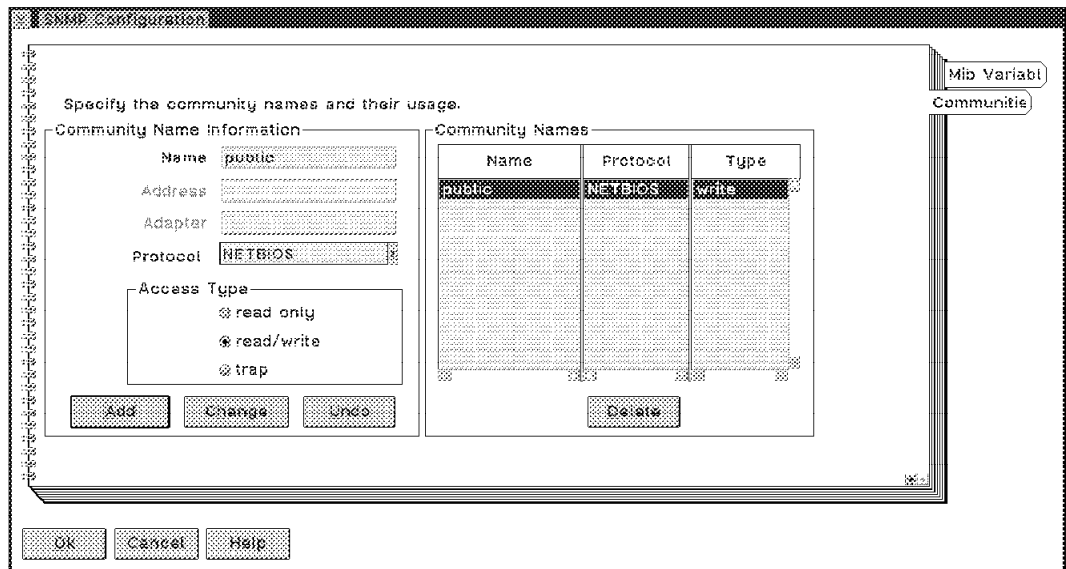


Figure 41. SNMP Community Names for Agents - NetBIOS Environment

2.6.2.2 Defining Your SNMP Trap Destinations (IP)

As a managed station, you will want to define where in the network you will be sending all your SNMP traps. On the same *Communities* page, you now select the **trap** radio button under *Access Type*. This will disable the *Network Mask* field, and will clear the the *Address* field. You should enter the IP address of the managing station that will receive your traps and then click on the **Add** button, and a new line item will appear under the *Community Names* list box on the right side of the window as shown in Figure 42.

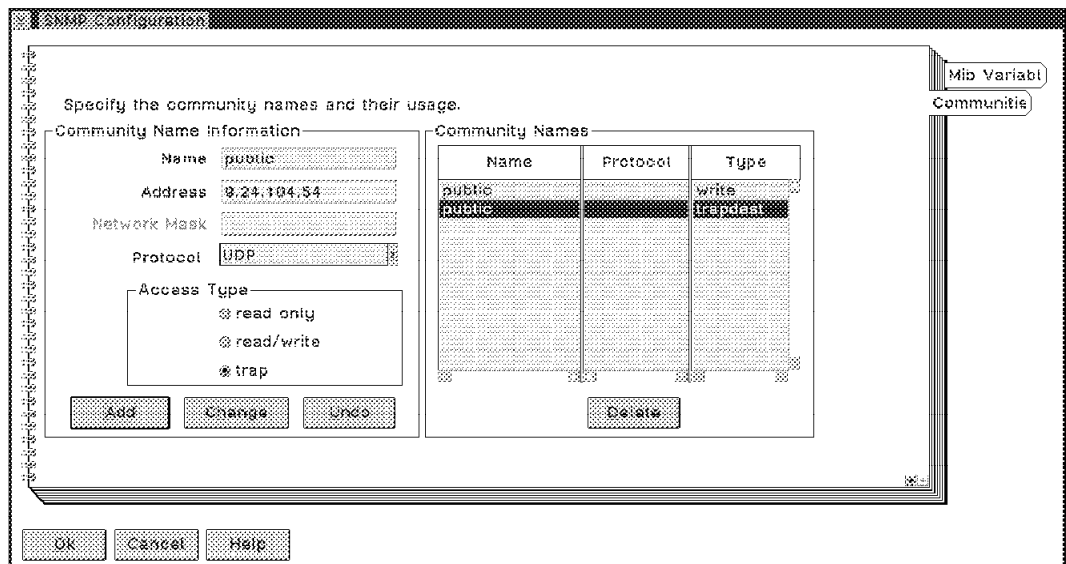


Figure 42. SNMP Trap Destinations (IP)

Once you have finished your agent configuration, press the **OK** button to continue with installation.

2.6.2.3 Defining Your SNMP Trap Destinations (NetBIOS)

As a managed station in a NetBIOS environment, you will want to define where in the network you will be sending all your SNMP traps. On the same *Communities* page, you now select the **trap** radio button under *Access Type*. This will disable the *Address* field, and will create an entry field for an *Adapter*. You should enter the MAC address or LAA of the managing station that will receive your traps and then click on the **Add** button. A new line item will appear under the *Community Names* list box on the right side of the window as shown in Figure 43.

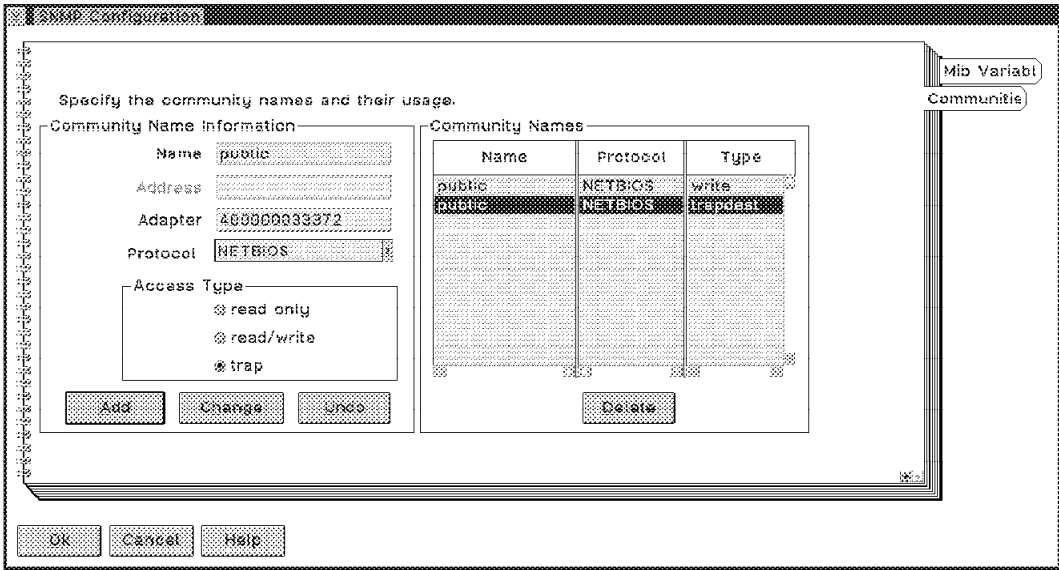


Figure 43. SNMP Trap Destinations (NetBIOS)

Note

As mentioned in the installation guide, do not click on the **Cancel** push button if some of the information for this panel is not yet available. You can always update it after you finish the install using the **OS/2 Agent Configuration** in the NetView for OS/2 folder.

The second window is accessed by selecting the **MIB Variables** tab on the SNMP Configuration window, as shown in Figure 44 on page 40. Note that the *Description* field is already filled with default information. The content of this description field should follow the SNMP guidelines for System Description (for example, management code running, on which machine type).

For Contact and Name, it is strongly recommended that you enter very meaningful names so that when something goes wrong and the network administrator needs to identify this machine, the descriptive fields will point exactly to the machine in question.

Be sure to enable Authentication Traps. This will ensure that the managing station will have the same Community Name as the managed station. Therefore, a first layer of security would be to ensure that managed systems cannot be managed by managing stations outside of their SNMP Community Name.

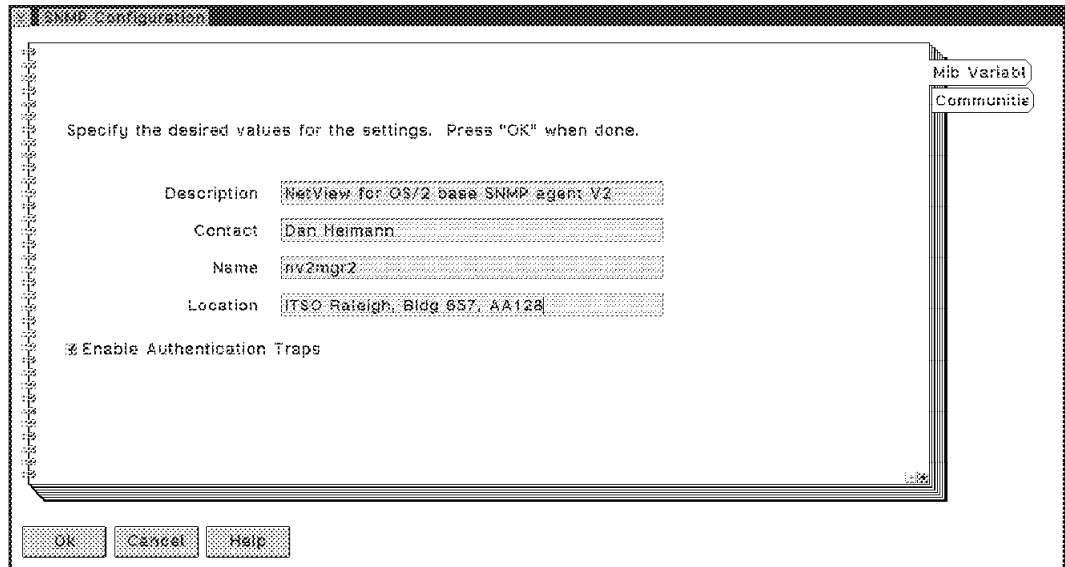


Figure 44. NetView for OS/2 SNMP Configuration for Agents

Once you have finished your agent configuration, press the **OK** button to continue with installation.

2.6.3 Installing and Configuring NetView for OS/2 Managing Systems

The installation of a *Managing* system is very much the same as for the *Agents* except for the following:

1. In the Install Directories window as shown in Figure 37 on page 34, you would press the **Select all** button to select all the management code, agent code, and documentation.
2. As you are also a managed machine, you can still only choose 1 protocol to communicate with the managing component in your machine.

Therefore, in Figure 39 on page 36, you can select only one radio button for Protocols.

3. After the two SNMP Configuration panels, you will be shown another Installation - Transports window similar to that displayed during the Agent code install. For the *Managing* station, you can now choose as many protocols as you have installed in order to manage the stations in your LAN environment.

Important for NetBIOS

On a *Managing* system running netbiosdiscovery, you may have to increase the value of your *Maximum Link Stations* parameter. If your Netbiosdiscovery daemon ends with an exit code of 70, then use LAPS to increase your *Maximum Link Stations* by a sufficient number such as 40. The *Maximum Link Stations* variable is found in the 802.2 configuration line item.

Also, to support MIB applications, we recommend the following changes in LAPS:

- Set *Full Buffer Datagram* = YES for the adapter card.
- Set *Transmit Buffer Size* = 4096 for NetBIOS support.

4. NetView for OS/2 will show you what installation-selected options you have made and will then copy all necessary programs to your hard disk. Once complete, a final window will appear stating: The requested components of IBM NetView for OS/2 are successfully installed. Elapsed time was hh:mm:ss.

2.6.3.1 Shutting Down and Rebooting

After you have installed all the program files, you need to shut down and reboot your system. When you do this, you will see that NetView for OS/2 has placed 2 icons in your Startup folder: one for OS/2 Agent Startup, and one for IBM NetView for OS/2 Startup. For an agent machine, just the OS/2 Agent Startup icon will be there. When you reboot a managing station, the NetView for OS/2 code will automatically startup. The first time that you reboot the system after installing the product you will see a NetView for OS/2 Setup window. It will not appear each time you reboot the system. You should see various messages letting you know that:

- The transport stack is being activated.
- Object registration for the NetView for OS/2 daemons was successful.
This is done using the SVADDOBJ command.
- Object registration for the LMU components was successful.

Once all the objects have been registered, the agents will be automatically started.

Finally, an SVSTART will be automatically submitted that will start all of the NetView for OS/2 daemons (processes).

Recommendation

It is recommended that you take the 2 startup icons out of the Startup folder so that you can control which component starts first. We dragged and dropped them to our NetView for OS/2 Icon View main folder. We recommend that the OS/2 Agents start first, followed by the the NetView for OS/2 daemons. You can do this by issuing the following commands sequentially:

1. START /MIN NV2AGTST
2. SVSTART

OR

1. Click on the *OS/2 agent startup* icon first, then
2. Click on the *IBM NetView for OS/2 Setup* icon.

Note: After you run the *IBM NetView for OS/2 Setup* program using this icon, NetView for OS/2 will change the icon name to *IBM NetView for OS/2 Startup*. To see what our Startup folder and NetView for OS/2 Icon View folders looked like, please see the following two figures:

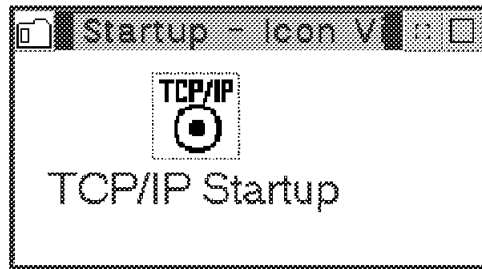


Figure 45. Startup Folder for the NV2MGR1 Machine without NetView for OS/2 Icons

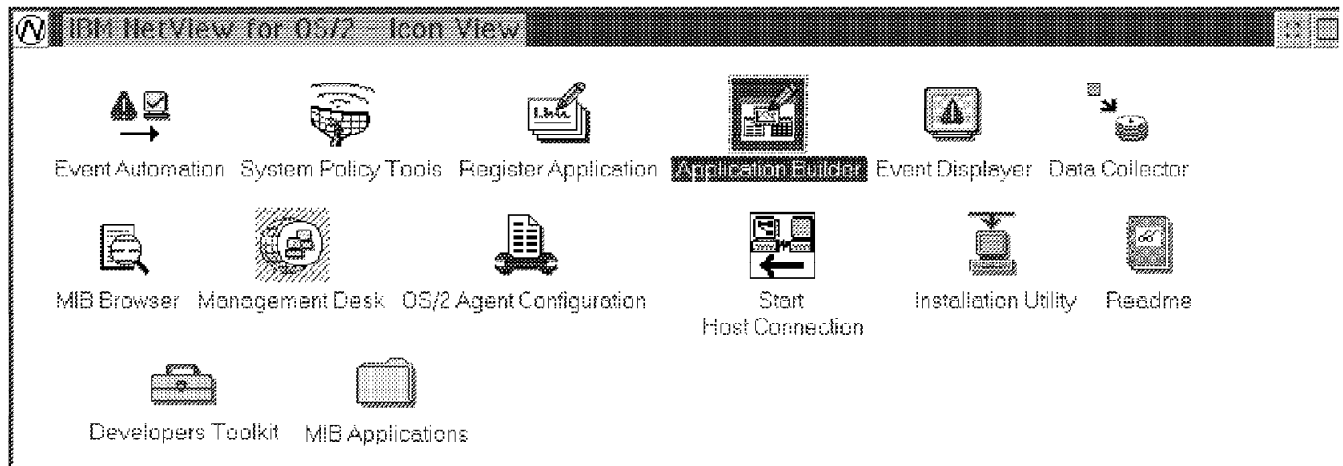


Figure 46. NetView for OS/2 Icon View Folder for the NV2MGR1 Machine

2.7 Starting Sequence for All Management Software

Now that you have all the prerequisite code installed, there is a specific order in which the code should start. The order depends upon what kind of machine it is being started on.

2.7.1 Starting a NetView for OS/2 Managed System with No LMU

1. Start TCP/IP, NetBIOS, IPX or AnyNet/2.
2. Start the NetView for OS/2 agent code.

2.7.2 Starting a NetView for OS/2 Managing System with No LMU

1. Start TCP/IP, NetBIOS, IPX or AnyNet/2.
2. Start the NetView for OS/2 agent code.
3. Start the NetView for OS/2 management code.

2.7.3 Starting a NetView for OS/2 Managed System with the LMU Proxy Agent

Note: This machine would also be an LMU managing system.

1. Start TCP/IP, NetBIOS, IPX or AnyNet/2.
2. Start the OS/2 LAN Requester or NetWare Requester.
3. Log on to your LAN.
4. Start Database 2/2.
5. Log on as the database administrator.
6. Start the NetView for OS/2 agent code on this LMU proxy agent station.
7. Start the LMU management code (LMUSTART) on this LMU proxy agent station.

2.7.4 Starting a NetView for OS/2 Managing System with the LMU Proxy Agent

Note: This machine would also be an LMU managing system.

1. Start TCP/IP or NetBIOS or IPX or AnyNet/2.
2. Start the OS/2 LAN Requester or NetWare Requester.
3. Log on to your LAN.
4. Start Database 2/2.
5. Log on as database administrator.
6. Start the NetView for OS/2 agent code on this LMU proxy agent station.
7. Start the LMU management code (LMUSTART) on this LMU proxy agent station.
8. Start the NetView for OS/2 management code

Please see the sample in Figure 47 on page 44 that shows the STARTUP.CMD file that we used on the NV2MGR1 machine. It was used as both a NetView for OS/2 managing station and an LMU managing station. It was also acting as an LMU proxy agent for downstream OS/2 and DOS/Windows LMU agent systems. The machine would start up using this procedure and then we would start the following programs manually in a specific order as follows:

1. Start the NetView for OS/2 agent code. It is imperative to have SNMPD start before starting the LMU proxy agent.
2. Issue an LMUSTART command. This would start the LMU managing system and the LMUSNMPD daemon (the LMU proxy agent).
3. Start the NetView for OS/2 management programs either through the icon or by issuing an SVSTART command.

You could include the above three steps in your STARTUP.CMD as well. We decided not to include them, because we were using the machine differently at each start up.

```

align-center.
/* Start Up Command File */
'Echo On'
/* Start the OS/2 LAN Requester */
'net start requester'
/* Start Communications Manager/2 with Host Connectivity and APPC */
'start c:\cmlib\cmstart.exe wtrmodel'
/* Start DB2/2 and logon as an Administrator (SYSADM) */
'startdbm'
'logon USERID /p:PASSWORD /V:local'
/* Login to the NetWare Server */
'c:\netware\login bank311/password'
/* Logon to the OS/2 LAN Server as Administrator for LMU/2 */
'logon lmumgr /P:xxxxxx /V:D /D:nvsrvdm'
Address CMD Exit 0

```

Figure 47. STARTUP.CMD File for NetView for OS/2 Managing Station with LMU Proxy

Chapter 3. NetView for OS/2 Functional Overview

In this chapter, we will show how the main functions of NetView for OS/2 work. The functions that we will show are:

- Discovery processes
- MIB Loader
- MIB Browser
- Application Builder and Executor
 - Application Registration
- Data Collector
 - Graphing MIB values
- Event Automation and Event Display
- NetView for OS/2 databases
- Host Connectivity

3.1 Discovery Processes

The discovery process is used to populate the All Systems folder with machines discovered in your network. A system of seed files and masks can limit discovery based on specified criteria. NetView for OS/2 discovers machines which use one or more of the TCP/IP, IPX, and NetBIOS protocols. Each of these discovery processes can be turned on and off as needed; however, all systems already discovered will remain in the All Systems folder until you reboot your system. Discovered systems can be grouped into user defined collections based on filtering criteria. In this section, we will cover the following topics:

- Describe how to start and stop the three types of discovery.
- Create a log file of nodes discovered.
- Use a seed file and a mask file to limit TCP/IP discovery.
- Set up a user-defined management collection.
- Create a custom filter.

3.1.1 Description of Discovery Process

The discovery process starts by default when you start NetView for OS/2. Discovery takes place using three different protocols: TCP/IP, IPX, and NetBIOS. You can check what discovery processes are running by typing `SVSTATUS` on a command line. This will list all the processes and their status. The three processes responsible for discovery of new systems are `tcpipdiscovery`, `netbiosdiscovery` and `netwdiscovery`. They can be started or stopped separately by typing in `SVSTART discoverytype` or `SVSTOP discoverytype` on the command line where *discoverytype* is one of the three discovery processes listed above.

Discovery can be limited using seed files and masks. Seed files can be created for TCP/IP and NetBIOS discovery. The TCP/IP seed file contains a list of IP addresses for individual systems and the NetBIOS discovery seed file contains segment numbers within an administrative domain on a local network. A mask file can be created for TCP/IP discovery to limit additional discovery.

User-defined collections can further subdivide the discovered systems into logical collections which can be easily managed. A good management strategy would be to divide all systems discovered into logical collections which would allow the network administrator quick and easy access to managed systems or groups of managed systems. There are many criteria to consider when organizing user collections. These criteria include:

- Logical - Grouped by location, infrastructure, configuration, hardware, operating system, or protocol.
- Availability - Key systems which handle important resources in the network like file servers, print servers, or printers.
- Status - Problem systems, degraded, failed, unknown or offline. Systems that may need attention could be a collection.
- Security - Systems which are monitored for security reasons.
- Usage - Most frequently accessed or common tasks performed.
- Other - User specific criteria based on needs of the administrator.

The final decision of the collection structure is up to the network administrator and may differ from one network to the next based on individual needs.

3.1.2 Using Log Files and Seed Files to Limit Discovery

To manage a collection of systems, you may want to discover certain known systems at startup, and then continue to discover other systems according to some specific masking criteria. Creating a seed file of systems to be discovered at startup can be a tedious task if there are a large number of systems as shown in Figure 48 on page 47.

The process of creating a seed file can be simplified by having the system create a file of all systems discovered. You can then edit this file to create a seed file with only the systems that you want to be discovered at startup. Then, by using a mask, you can further limit discovery to only certain groups of systems.

The advantage of a seed file is that these systems will be discovered regardless of how your mask is set. The resulting effect will be an All Systems folder containing only those systems that you are managing and not potentially thousands that you don't care about. In addition, there may be cases where you want another system to take over the management of the environment, and you can set up a script to pass the seed file over to another manager. It is also possible that a manager on another platform (for example, NetView for AIX), may have some use for that information.

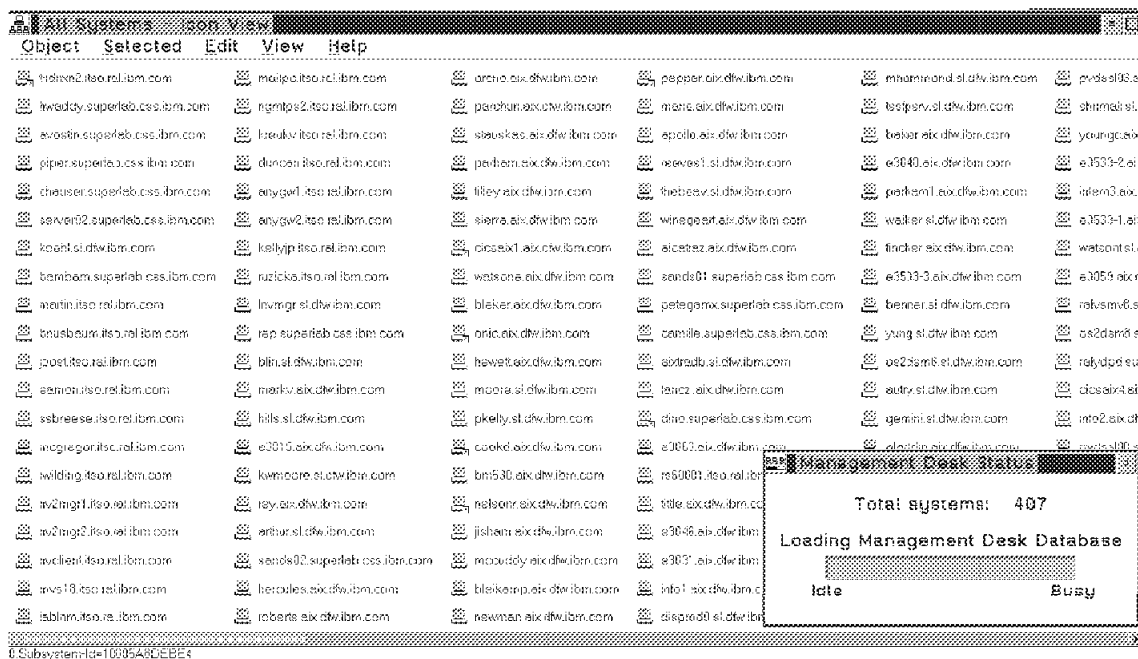


Figure 48. All Systems Folder with All Systems Discovered

We would like to create an All Systems folder containing the following 13 systems:

1. bambam.superlab.css.ibm.com
2. alcatraz.aix.dfw.ibm.com
3. e3054.aix.dfw.ibm.com
4. nvclient.itso.ral.ibm.com
5. nv2mgr1.itso.ral.ibm.com
6. nv2mgr2.itso.ral.ibm.com
7. mvs18.itso.ral.ibm.com
8. anygw1.itso.ral.ibm.com
9. mailpc.itso.ral.ibm.com
10. eamon.itso.ral.ibm.com
11. anygw2.itso.ral.ibm.com
12. joost.itso.ral.ibm.com
13. kellyjp.itso.ral.ibm.com

The seed file will contain 10 systems from the 9.24.104 subnet and 3 systems which are outside of this subnet. The mask file will ensure that other systems discovered will only be in the 9.24.104 subnet. To facilitate the creation of a seed file, we first created a log file with no masking. We discovered over 400 systems. Now, we can manually edit this LOG file and delete those entries we do not want. The resultant file is our seed file. We gave it the name SEEDLIST.LOG.

3.1.2.1 Creating a Log File

The log file of all the discovered systems is called NODENAME.LOG and is located in the \anv2\etc\ directory. This name cannot be changed. If you want to save a log of systems found, you should copy NODENAME.LOG to a different file name. The steps for creating a log of all systems discovered is as follows:

1. Stop TCP/IP discovery by typing:
SVSTOP tcpipdiscovery

2. Edit the \ANV2\ETC\LRF\LNTCPIP.LRF file to include an -L in the third field of the second line. Each field is separated by colons.

Default value is:

```
tcipdiscovery:LNTCPIP.EXE:OVs_CMOT_TCPIP:
OVs_YES_START:postmaster,topology,agentdiscovery,:OVs_DAEMON::
```

Change it to:

```
tcipdiscovery:LNTCPIP.EXE:OVs_CMOT_TCPIP:
OVs_YES_START:postmaster,topology,agentdiscovery:-L:OVs_DAEMON::
```

3. Save the file LNTCPIP.LRF in the \anv2\etc\lrf\ directory.
4. Run the facility to add a new object to to the LNTCPIP.LRF file by typing:
SVADDOBJ LNTCPIP.LRF
5. Start TCP/IP discovery by typing:
SVSTART tcipdiscovery

6. The discovery process will start and new systems will be added to the All Systems folder. The NODENAME.LOG file will be created when TCP/IP discovery is stopped. Stop TCP/IP discovery by typing:

```
SVSTOP tcipdiscovery
```

and with an editor look at the \ANV2\ETC\NODENAME.LOG to see its contents. You should have a list of TCP/IP addresses which were found during the discovery process. Anything after the # sign is a comment. Anytime you start and stop TCP/IP discovery, the NODENAME.LOG file is changed. To keep the current copy of NODENAME.LOG, copy it to another file name or stop the LOG option by removing the -L from the \ANV2\ETC\LRF\LNTCPIP.LRF file and run svaddobj lntcip.lrf.

The same principle follows for NetBIOS and IPX discovery in terms of the -L option, and the nodelist file.

3.1.2.2 Creating and Using the Seed File for TCP/IP

Once you have created a NODENAME.LOG file you can create your own seed file by copying NODENAME.LOG to another file, editing this file and using this new file as your seed file, or you can create your own seed file directly. We used the NODENAME.LOG file to create our seed file.

1. Copy the NODENAME.LOG file to a file called SEEDLIST.LOG by typing:
COPY \ANV2\ETC\NODENAME.LOG \ANV2\ETC\SEEDLIST.LOG
2. Edit the SEEDLIST.LOG file to include only those machines you wish to discover at startup. You are allowed to use a # to comment out unwanted addresses. Our resultant seed file is as follows:

```
bambam.superlab.css.ibm.com      # 9.24.96.76
alcatraz.aix.dfw.ibm.com         # 9.19.129.193
e3054.aix.dfw.ibm.com           # 9.19.129.182
#
nvclient.itso.ral.ibm.com        # 9.24.104.68
nv2mgr1.itso.ral.ibm.com         # 9.24.104.54
nv2mgr2.itso.ral.ibm.com         # 9.24.104.55
mvs18.itso.ral.ibm.com           # 9.24.104.74
anygw1.itso.ral.ibm.com          # 9.24.104.117
mailpc.itso.ral.ibm.com          # 9.24.104.101
eamon.itso.ral.ibm.com           # 9.24.104.18
```

```
anygw2.itso.ral.ibm.com      # 9.24.104.118
joost.itso.ral.ibm.com      # 9.24.104.17
kellyjp.itso.ral.ibm.com    # 9.24.104.122
```

3. Save this file as SEEDLIST.LOG or any other valid name you choose. Unlike the NODENAME.LOG, you can specify any valid DOS name you like.
4. Edit the \anv2\etc\lrf\lntcpip.lrf file:

```
tcpipdiscovery:LNTCPIP.EXE:OVs_CMOT_TCPIP:
OVs_YES_START:postmaster,topology,agentdiscovery:-L:OVs_DAEMON::
```

Change it to:

```
tcpipdiscovery:LNTCPIP.EXE:OVs_CMOT_TCPIP:
OVs_YES_START:postmaster,topology,agentdiscovery:-s \anv2\etc\seedlist.log,-L:OVs_DAEMON::
```

Note

The -s \anv2\etc\seedlist.log and -L must be separated by a comma or a space. Do not leave spaces anywhere else.

Do not use the ':' in the path because it is used as a delimiter. You will get an error when you run SVADDOBJ.

5. Save the file in the \anv2\etc\lrf directory
6. Stop TCP/IP discovery by typing:
SVSTOP tcpipdiscovery
7. Add the lntcpip.lrf file by typing:
SVADDOBJ LNTCPIP.LRF
8. Start TCP/IP discovery by typing:
SVSTART tcpipdiscovery

3.1.2.3 Limiting Discovery of Managed Systems

The last step in customizing the All Systems folder to contain only the systems that we want to see, is to create a mask file that adds discovered systems to the folder according to our specified criteria. This file is called LNTCPIP.MSK, and it is located in the \anv2\etc\ directory. You can specify a list of valid addresses or use the wildcard character (*) to mask out certain addresses. The default mask is *.*.*.*. That is, all systems are added to the folder. To change the mask, edit the LNTCPIP.MSK file. Once changed, the new mask will take effect, however, all systems already discovered remain in the discovery database. We changed the file as follows:

The system default value is :

```
# TCPIP Discovery Scope Mask File
*.*.*.*
```

In our example, we changed it to:

```
# TCPIP Discovery Scope Mask File
# *.*.*.*
9.24.104.*
```

Once you save the file the new mask should take effect.

To have all the changes take effect and the discovery database cleared, shut down NetView for OS/2 and reboot the system.

The new All Systems folder created as a result of using the seed file is shown in Figure 49. Note that the total systems went from 407 to 13.

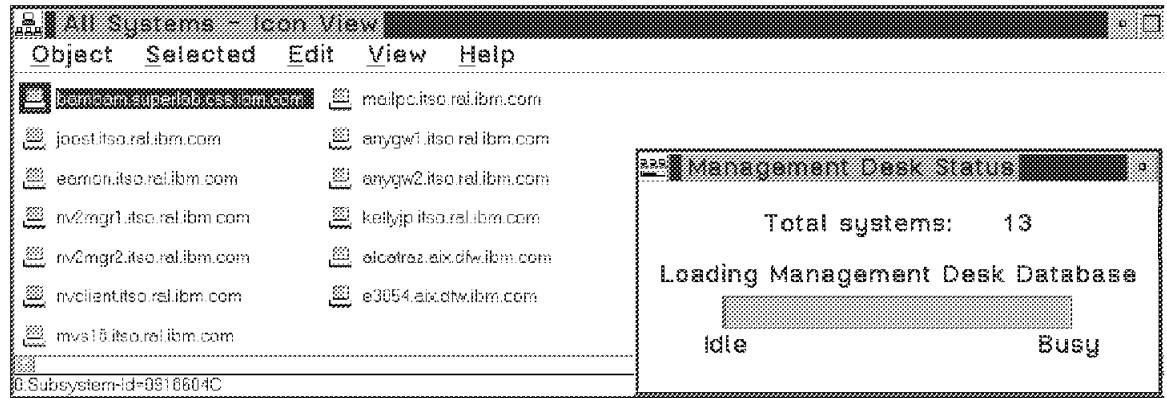


Figure 49. All Systems Folder Created from SEED File

3.1.3 How to Set Up a Management Collection

The All Systems collection contains all discovered systems in your network. This All Systems collection can grow to over thousands of systems, and at that size it can become very difficult to manage. You can create your own folders or collections to break up that large one into more manageable collections. Once a system is discovered, it cannot be removed from the All Systems collection. In your own collection, no such restrictions apply. To better organize the discovered systems, you can create collections based on 4 primary filters and custom filters which you create. The 4 basic filters are:

1. *All Systems Filter.* A collection with this filter is populated with everything in the all systems collection automatically; however, systems can be deleted from this collection.
2. *No Systems Filter.* An empty collection where the only way to populate it is to drag and drop systems into it. This one is useful for creating collections based on departments, buildings, or other common things, where there is no filtering criteria for placing the systems in automatically.
3. *NetWare Filter.* A collection of systems which have an IPX stack are placed into this collection. You can create a collection with only NetWare machines according to network address.
4. *Attribute Filter* This filter allows filtering based on:
 - a. Short name - The name of the machine defined during configuration (icon name).
 - b. Host name - The IP address of host systems can use wildcard characters.
 - c. NetWare node name - The MAC address of the adapter used by a NetWare system.
 - d. SNMP system description - The character text for describing this system.
 - e. SNMP system contact - The text for the person responsible for this system.

- f. SNMP system location - The text location for this system.
- g. SNMP system object identifier - The object ID that NetView for OS/2 knows this system by.
- h. Status:
 - 1) Normal - No errors have been found.
 - 2) Degraded - Some resources are not responding but the system is running.
 - 3) Failed - The system is not responding due to an error.
 - 4) Unknown - The system status is unknown.
 - 5) Offline - The system is turned off or not accessible.
- i. Resource - The MIB resource type:
 - IP host (1.3.18.0.0.3315.65.3.6)
 - IP gateway (1.3.18.0.0.3315.65.3.7)
 - NetWare machine (1.3.18.0.0.3315.65.3.2)
 - NetWare server (1.3.18.0.0.3315.65.3.4)
 - NetWare workstation (1.3.18.0.0.3315.65.3.5)

Collections can be created based on these 4 filters which can be customized to suit the user's needs. Systems can be isolated based on:

- The need for some action. They contain systems which are degraded, failed, offline or unknown. They can be set up to be updated automatically, removing and inserting systems as their status changes.
- TCP/IP systems can be isolated by creating a collection with the host name in the attribute filter set to *.
- NetWare collections can be created by setting the network address to a segment in your network in the network filter.
- User populated collection created by dragging and dropping systems into a collection with the No Systems filter.

An example of a collection based on status is shown in Figure 60 on page 58. The following steps were taken to create a collection with status Offline. The **All Systems Collection** shown in Figure 50 on page 52 has one system with the status of offline highlighted.

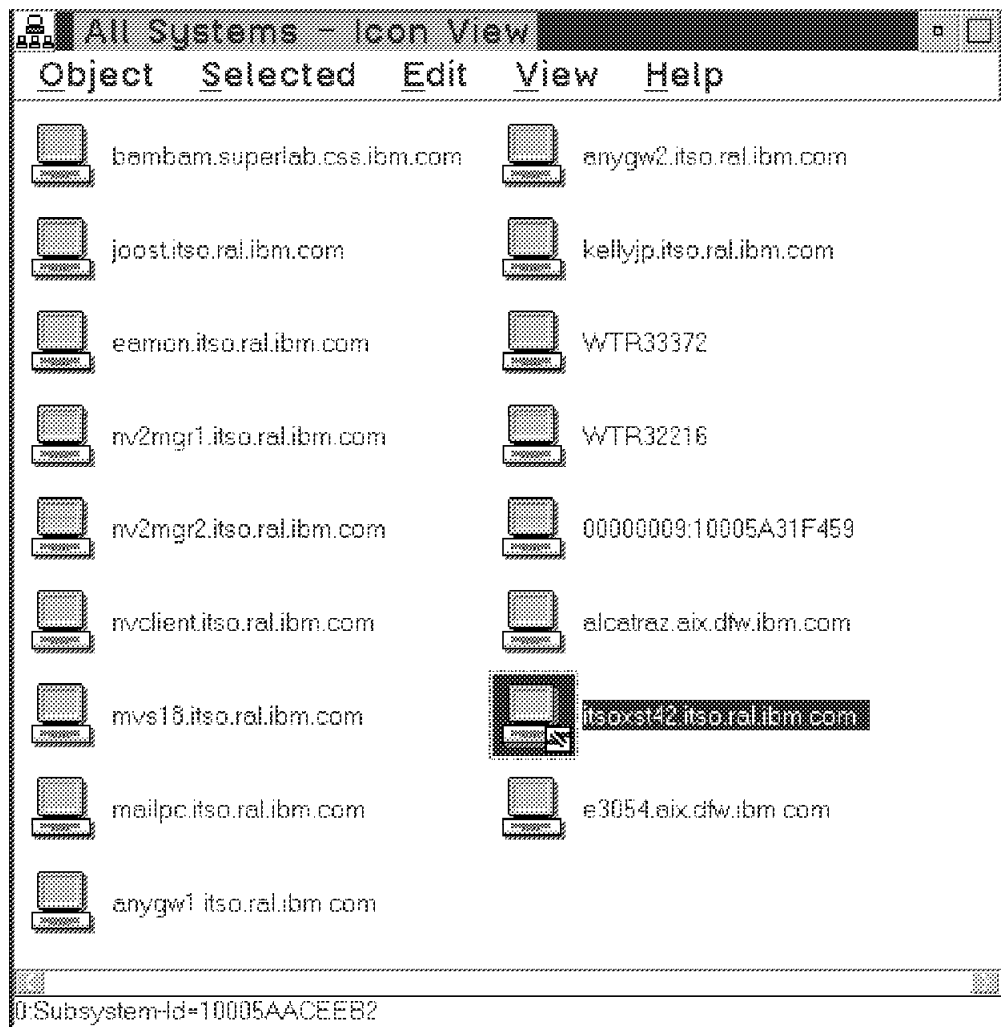


Figure 50. Example of an Offline System

1. Drag and drop the Template icon in the Management Collections window with the right mouse button to any open area in the window. See Figure 51. The Configuration Notebook for a new collection will be displayed.



Figure 51. Management Collection Template

2. The Configuration Notebook of the new collection is shown in Figure 52 on page 53. You must enter a unique name in the Name field. We called our collection **Offline Systems**. This name does not have to be the same as the icon name.

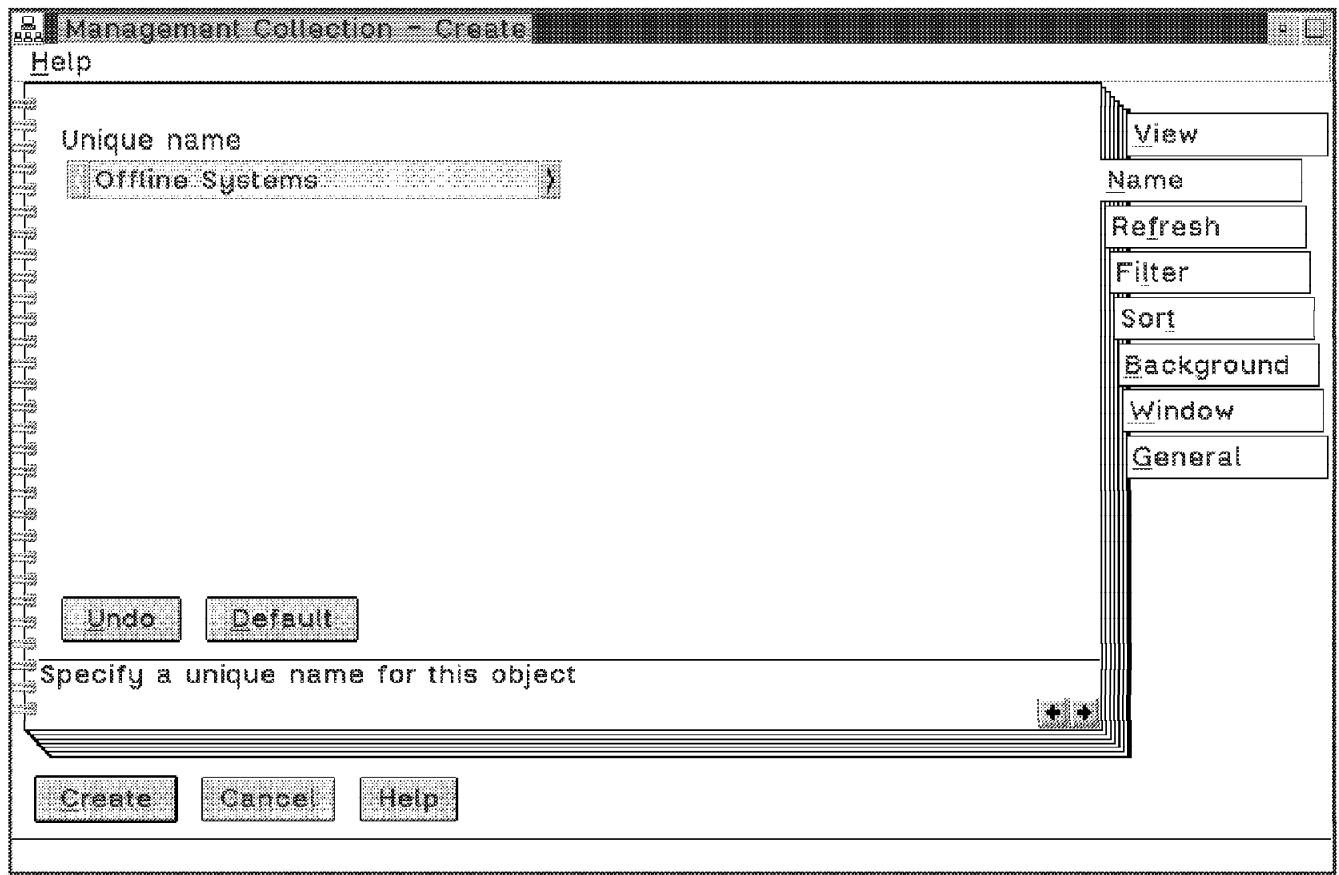


Figure 52. Management Notebook for Custom Collections

3. We want the collection to update the systems in its folder automatically. When a system is no longer Offline, we do not want it in this folder. To automatically update the collection after a certain time interval click on the Refresh tab. Then click on the **On** radio button in the *Timed refresh* box and enter a value in minutes for the time interval between refreshes. The values can be between 5 and 99 minutes. We set it to 5 minutes. See Figure 53 on page 54.

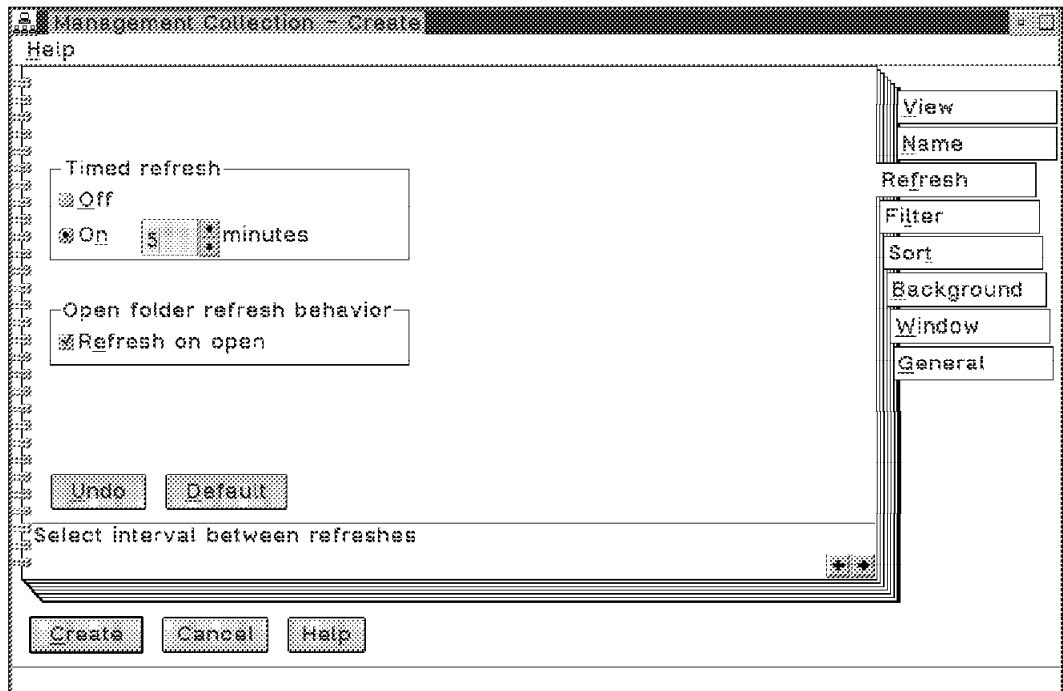


Figure 53. Setting the Timed Refresh Interval

4. To create a filter for this collection select the Filter tab, as shown in Figure 54. Click on the Attribute filter using the right mouse button and select **Create another...** from the menu.

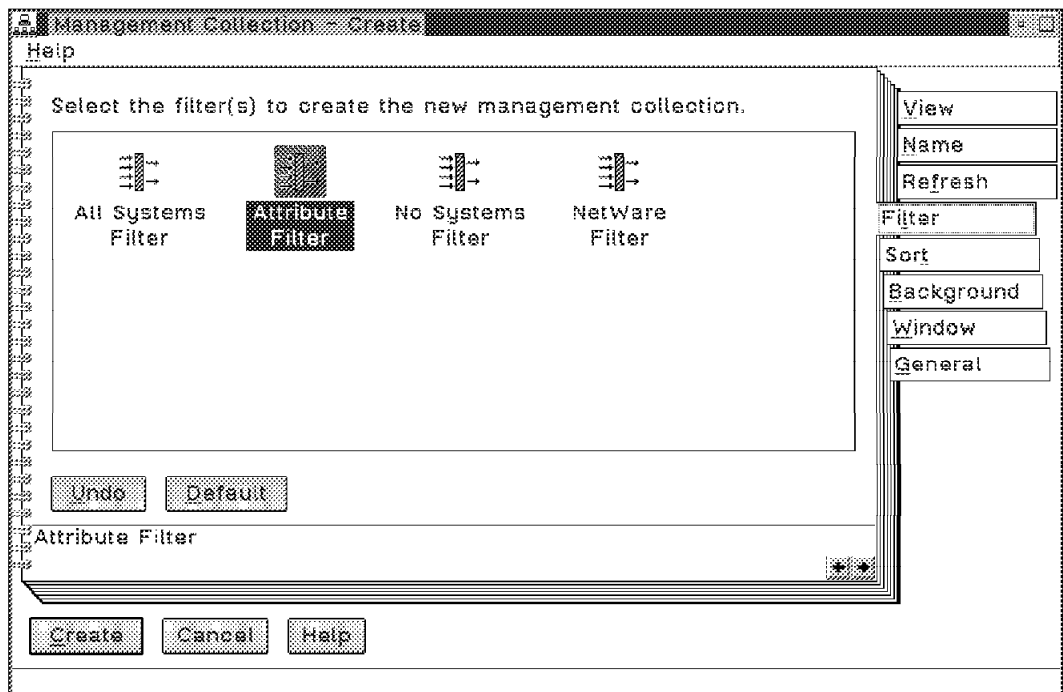


Figure 54. Creating a New Attribute Filter

5. Select the Name tab and enter a unique name for the filter. We recommend creating a convention for naming filters to correspond to the collection name

so that it is easy to see which filters belong to a collection. See Figure 55 on page 55. We named our filter Offline Systems Filter.

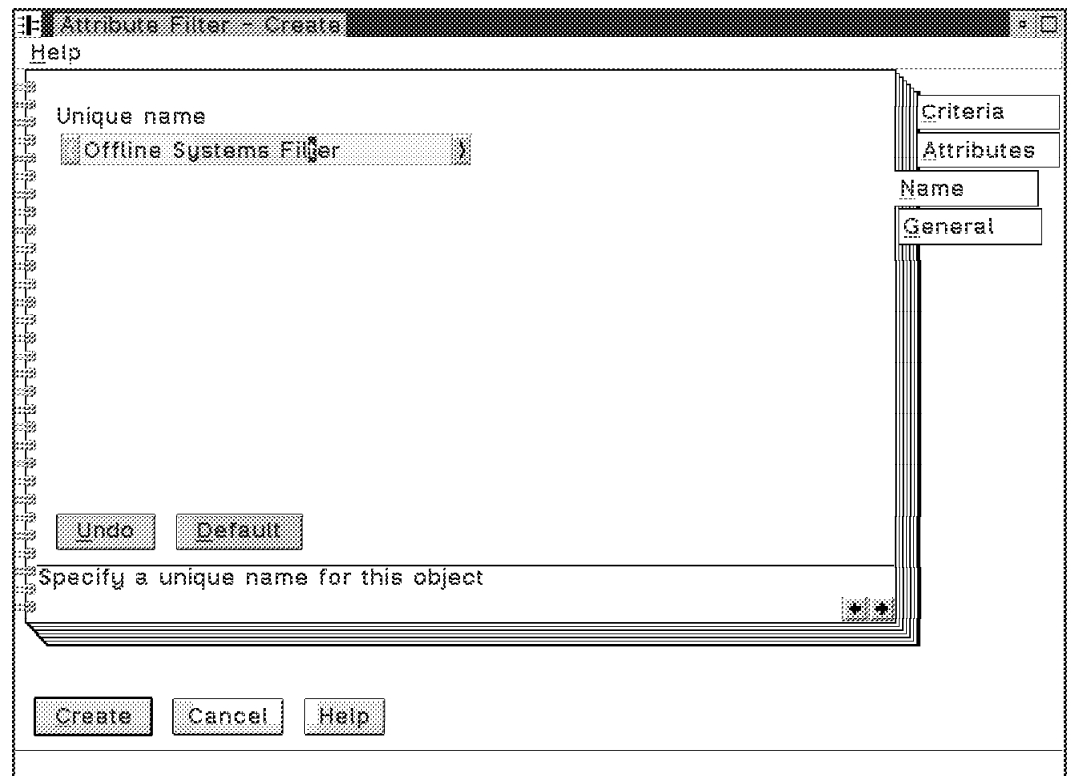


Figure 55. Entering a Name for the Filter

6. Click on the Attributes tab and expand the options in the Status field. A list of possible status choices appears. Scroll up and down until you find Offline and select it. See Figure 56 on page 56.

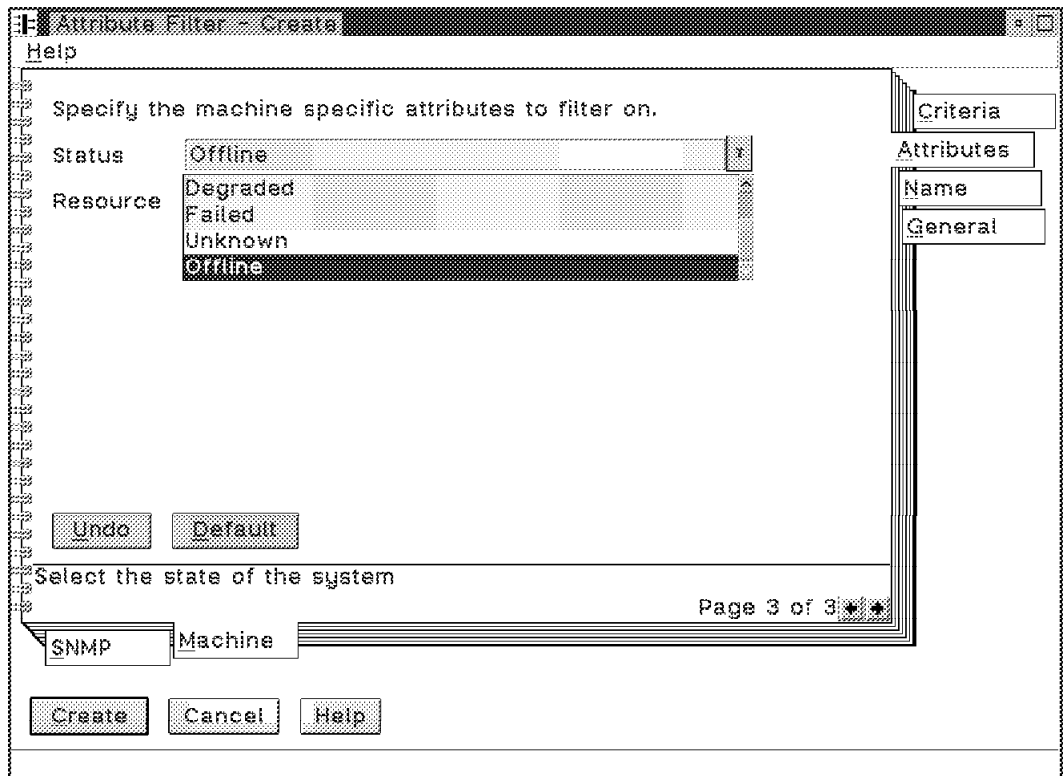


Figure 56. Setting the Filter Attributes

- Click on the General tab. Enter a filter name for the icon in the title box. We typed in Offline Systems Filter. See Figure 57.

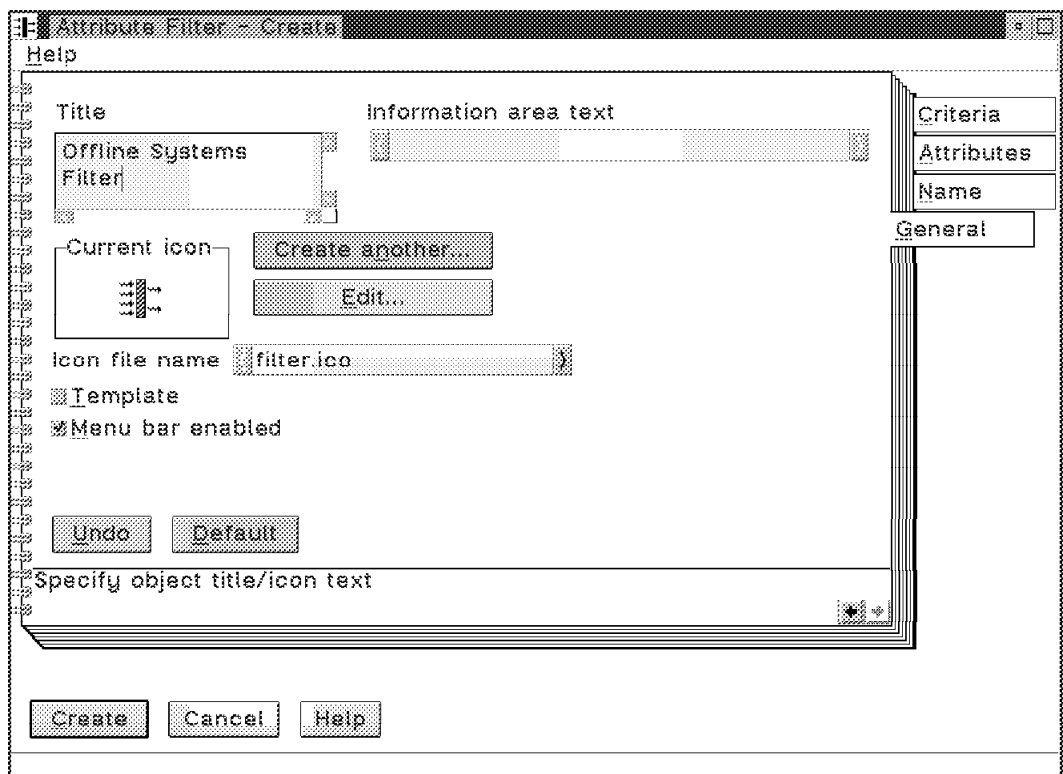


Figure 57. Giving the Filter Icon a Name

8. Click on the Create button at the bottom.
9. Click on the OK button to save the setting for the filter. See Figure 58.

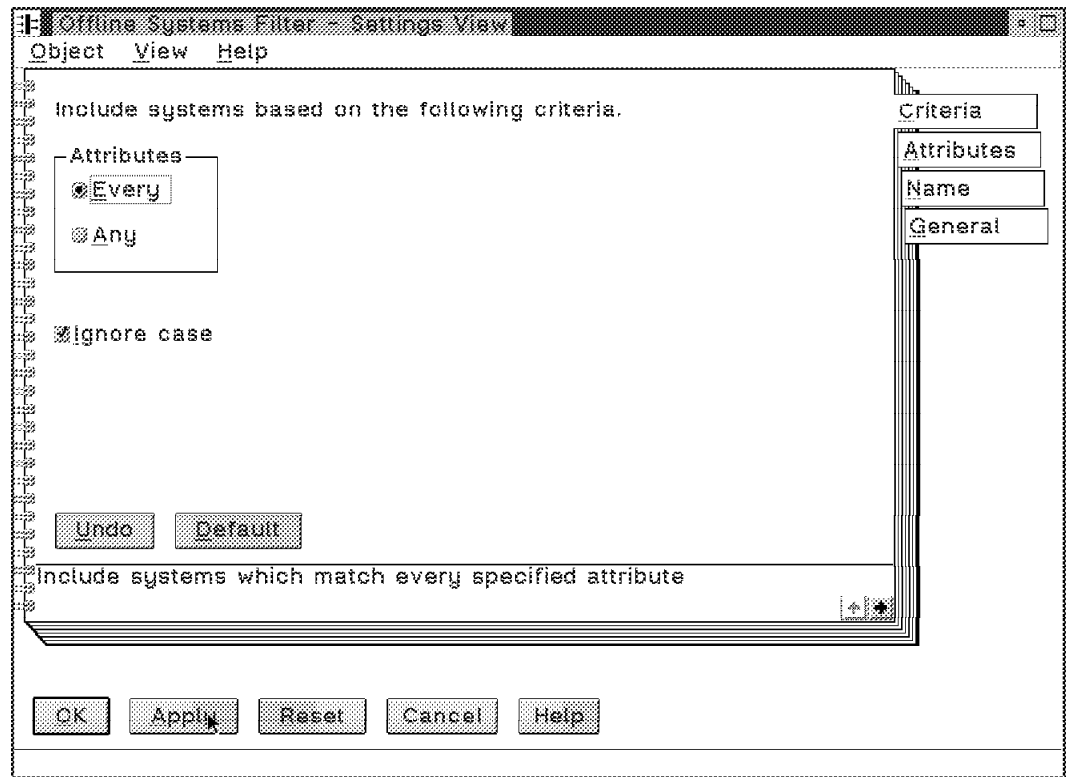


Figure 58. Press Apply

10. If you still have the original Attribute Filter window open, close it by clicking on the Cancel button.
11. Click on the Create button to create the new management collection. A new collection called Offline Systems now appears in the Management Collections - Icon view. See Figure 59.

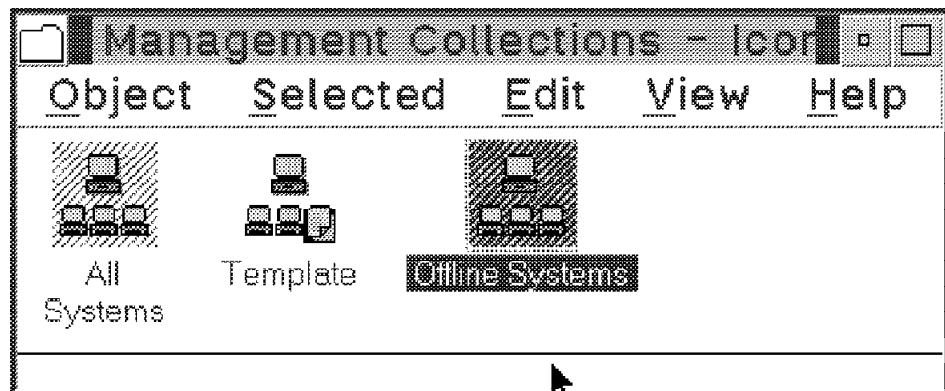


Figure 59. Setting the Management Collection Name

12. Open the Offline Systems collection by double clicking on the icon. In our case we had only one system offline and it appeared in the Offline Systems collection as shown in Figure 60 on page 58. This window automatically adds all of the offline systems and removes them when they change state

back to online. The screen is refreshed every five minutes so it may take some time for a system to appear in the collection folder once it goes offline.

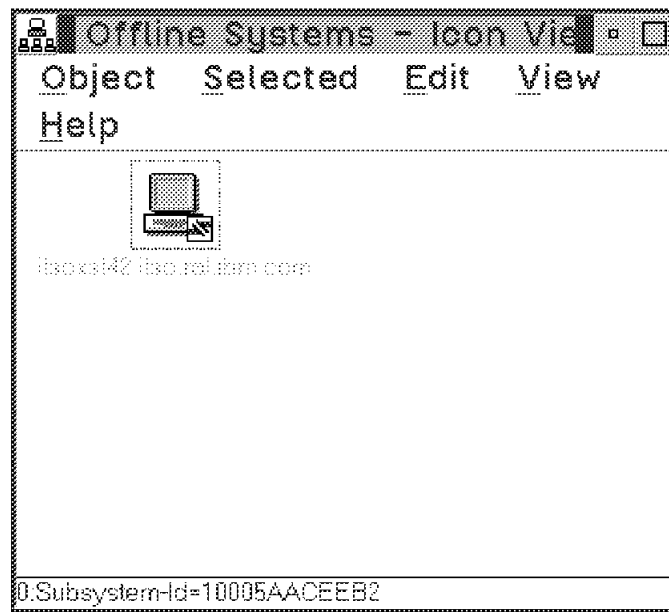


Figure 60. New Folder of Offline Systems

The above process can be modified to create any collection. Figure 61 on page 59 is a sample of possible collections created using different filters. Starting from the upper left corner going counterclockwise the collections are:

- *tcpip* - All systems having a TCP/IP address. Attribute filters for the host name are set to *.
- *NetWare* - All NetWare systems on network address 00000009. The NetWare filter with a network address set to 00000009.
- *LMU2* - LMU/2 managed systems. The No Systems filter was used and all entries were dropped in manually.
- *unknown* - Systems with a status of unknown. The Attribute filter was set to status unknown.
- *degraded* - Systems with a status of degraded. The Attribute filter was set to status degraded.
- *offline* - Systems with status offline. See 3.1.3, "How to Set Up a Management Collection" on page 50 for steps on how to create this collection.

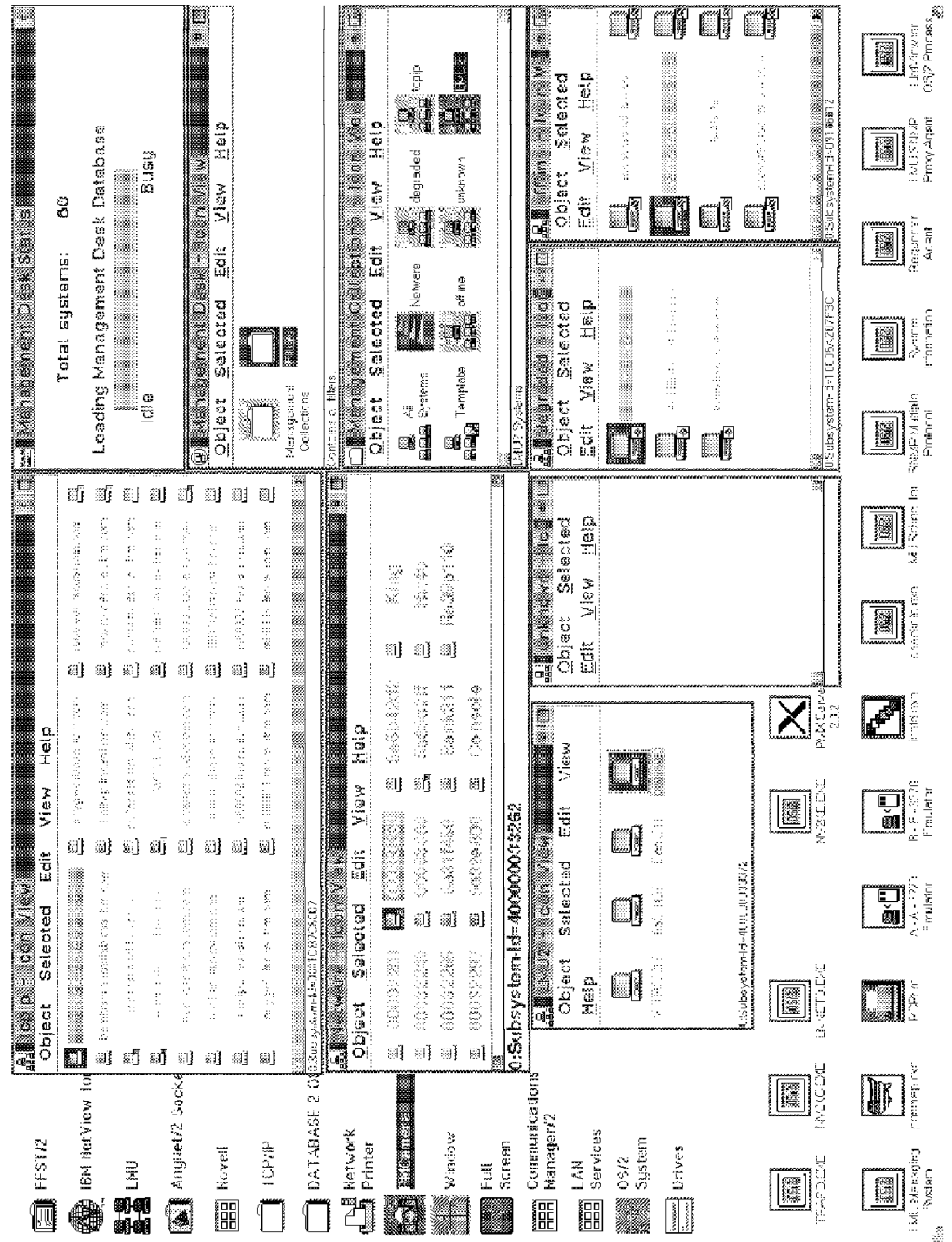


Figure 61. Sample Management Collections

3.2 MIB Loader

You can use the NetView for OS/2 MIB Loader (LOADMIB command) to append your SNMP device-specific MIB to the existing standard MIBs supplied with NetView for OS/2. The ITSO at Raleigh has a CISCO router. We will show how to first get the MIB from a RISC System/6000 that is attached to the router using the TCP/IP File Transfer Program (FTP), and then show how to load a MIB using the LOADMIB command. Finally, we will show how you can then use the MIB Browser and other NetView for OS/2 applications to monitor and control this device.

3.2.1 Obtaining the CISCO Router MIB via FTP

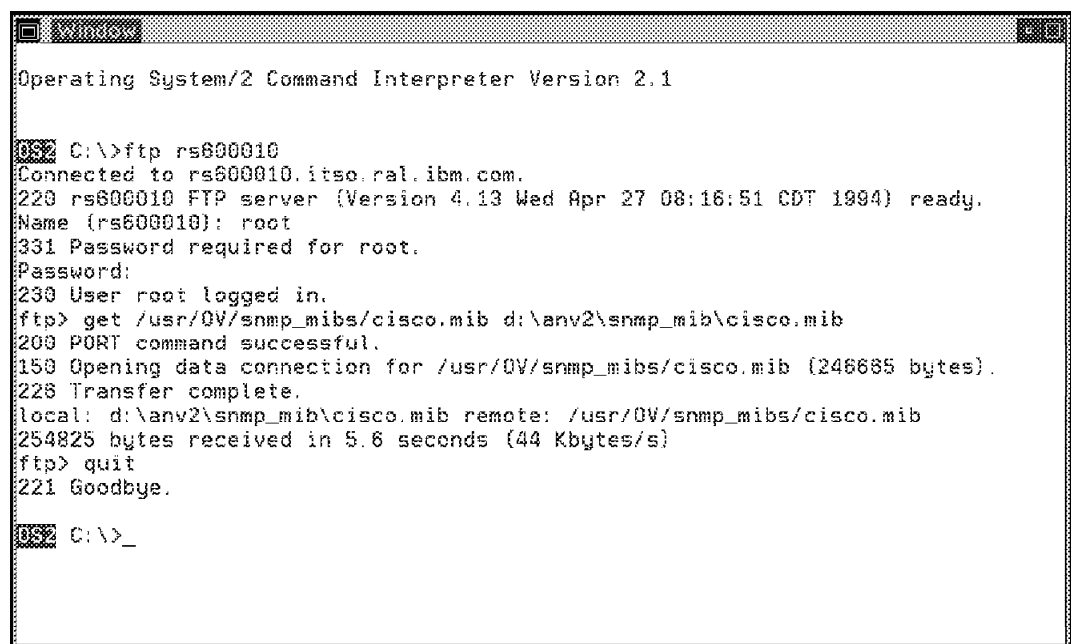
We followed these steps to get the MIB from the CISCO router:

1. We used the TCP/IP FTP program to access the RS/6000 where the MIB was located.
2. We then logged in to the Root user ID.
3. Then, we issued the appropriate GET command to copy the MIB to the required OS/2 directory:

```
GET /usr/OV/snmp_mibs/cisco.mib d:\anv2\snmp_mib\cisco.mib
```

4. Finally, we quit the FTP session to return to an OS/2 command prompt.

The OS/2 window shown in Figure 62 illustrates the whole process:



```
Operating System/2 Command Interpreter Version 2.1

C:\>ftp rs600010
Connected to rs600010.itso.ral.ibm.com.
220 rs600010 FTP server (Version 4.13 Wed Apr 27 08:16:51 CDT 1994) ready.
Name (rs600010): root
331 Password required for root.
Password:
230 User root logged in.
ftp> get /usr/OV/snmp_mibs/cisco.mib d:\anv2\snmp_mib\cisco.mib
200 PORT command successful.
150 Opening data connection for /usr/OV/snmp_mibs/cisco.mib (246685 bytes).
226 Transfer complete.
local: d:\anv2\snmp_mib\cisco.mib remote: /usr/OV/snmp_mibs/cisco.mib
254825 bytes received in 5.6 seconds (44 Kbytes/s)
ftp> quit
221 Goodbye.

C:\>_
```

Figure 62. Panel Showing Commands Needed to Get CISCO MIB

Acquiring MIBs

The MIB for the CISCO router came with the hardware. If you require an enterprise-specific MIB for an SNMP device, please contact the vendor for a copy of their MIB.

3.2.2 Loading the CISCO Router MIB

As you can see from Figure 62 on page 60, we placed the CISCO.MIB file in the D:\ANV2\SNMP_MIB directory. It must be there in order to use the LOADMIB command. At the OS/2 command prompt, enter the following NetView for OS/2 command:

```
LOADMIB -load cisco.mib
```

You would expect to see output similar to that shown in Figure 63.

```
mibfile to load: d:\ANV2\SNMP_MIB\cisco.mib
start cp=rfc1213-MIB-II
start cp=rfc1229-GINTF
start cp=rfc1230-802.4
start cp=rfc1231-802.5
start cp=rfc1232-DS1
start cp=rfc1233-DS3
start cp=rfc1243-APPLE
start cp=rfc1253-OSPF
start cp=rfc1269-BGP
start cp=rfc1271-RMON
start cp=rfc1284-ETHER
start cp=rfc1285-FDDI
start cp=rfc1286-BRIDGE
start cp=rfc1289-DECNET
start cp=rfc1304-SIP
start cp=rfc1315-FRAME
start cp=rfc1316-CHAR
start cp=rfc1317-RS232
start cp=rfc1318-PARALL
start cp=ibm.mib
start cp=ibm-alert.mib
start cp=ibm-nv6ksubagent.mib
start cp=ibm-6611-v1r1.1.mib
start cp=lmu2.mib
start cp=ibmsia6k.mib
start cp=lamib.mib
start cp=lsamib.mib
start cp=ibmsia.mib
start cp=host_s.mib
```

Figure 63. Output from Running LOADMIB Command

Now that the CISCO MIB has been successfully loaded, you can use the MIB Browser to monitor and control it as shown in 3.3.1, “Browsing the CISCO Router MIB” on page 62.

3.2.3 Acquiring MIBs from Other Devices

As you acquire your own enterprise-specific SNMP devices, you can load their MIBs into the NetView for OS/2 system MIB using a method similar to that described above. Just copy the MIB into the D:\ANV2\SNMP_MIB directory before executing the LOADMIB command. These MIB files are normally in binary format even though you can browse them. However, do not try to edit or change them.

3.3 MIB Browser

Now that you have loaded your enterprise-specific MIBs into the system and you have access to all the standard MIB-I and MIB-II variables supplied with NetView for OS/2, you will want to start to monitor and control these devices. You can use the MIB Browser to query variables and set the value of your MIB objects. If, when browsing the MIB, you do not understand what the variable is, then just click on the **Describe** push button and a definition of what this MIB variable is will be presented. You can also graph these MIB variables if the data type is an Integer, Counter or Gauge as shown in 3.3.2, "Graphing MIB Variables through the Browser" on page 64.

3.3.1 Browsing the CISCO Router MIB

After clicking on the NetView for OS/2 main icon, you can choose the MIB Browser icon to access the MIB Browser main panel. You can also open the Management Desk, discover and select the router, then click on the right mouse button to display a pop-up menu containing the MIB Browser. We chose the latter method because the address field was then automatically filled in with the name of our router, which was ciscoe as shown in Figure 64 on page 63. Note that the community name was not filled in, so the default was used for this system.

As an example of querying the CISCO router, we stepped through the following hierarchy to get to a MIB variable named chassis to find out what equipment was installed on the machine:

```
* private
  enterprises
    cisco
      temporary
        chassis
```

After getting down to the chassis level, we pressed the **Start Query** push button and the resulting report is shown in Figure 64 on page 63. From this report, we can see the following items:

- Chassis Type - C4000
- Processor RAM - 16MB
- Three adapter cards installed
 - Dual port serial interface
 - Single port token-ring
 - Dual port Ethernet

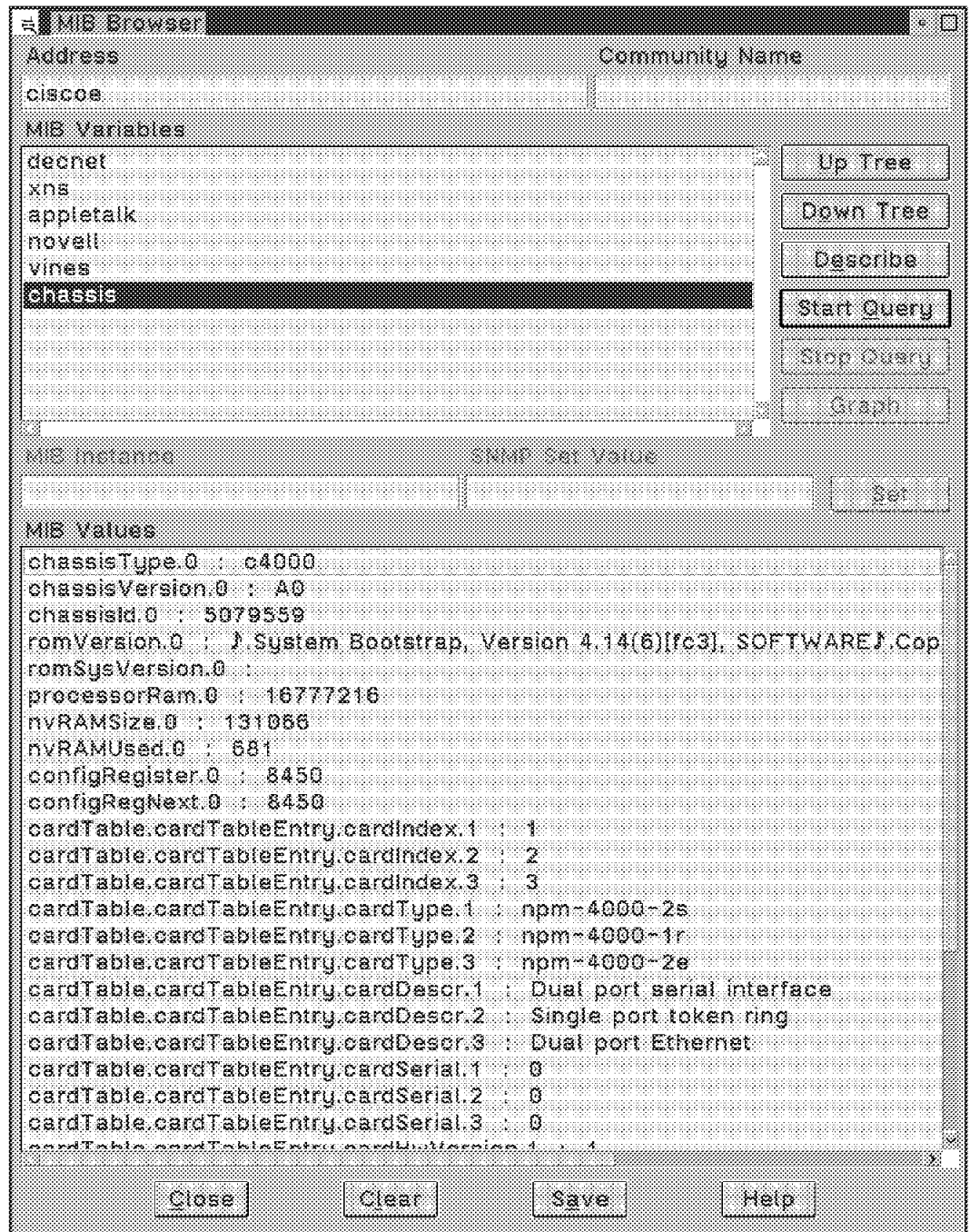


Figure 64. MIB Browser Window Showing Results from Query

You can also save this report in a file by pressing the **Save** push button and then viewing the d:\anv2\log\browser.sav file where the report is stored by default as shown in Figure 65 on page 64.

```

Address : 9.67.32.4

Community name :

MIB objects : chassis

MIB instance :

SNMP set value :

MIB values :
chassisType.0 : c4000
chassisVersion.0 : A0
chassisId.0 : 5079559
romVersion.0 : System Bootstrap, Version 4.14(6) fc3 , SOFTWARE . Copyright (c)
romSysVersion.0 : 1986-1994 by cisco Systems
processorRam.0 : 16777216
nvRAMSize.0 : 131066
nvRAMUsed.0 : 721
configRegister.0 : 8450
configRegNext.0 : 8450
cardTable.cardTableEntry.cardIndex.1 : 1
cardTable.cardTableEntry.cardIndex.2 : 2
cardTable.cardTableEntry.cardIndex.3 : 3
cardTable.cardTableEntry.cardType.1 : npm-4000-2s
cardTable.cardTableEntry.cardType.2 : npm-4000-1r
cardTable.cardTableEntry.cardType.3 : npm-4000-2e
cardTable.cardTableEntry.cardDescr.1 : Dual port serial interface
cardTable.cardTableEntry.cardDescr.2 : Single port token ring
cardTable.cardTableEntry.cardDescr.3 : Dual port Ethernet
cardTable.cardTableEntry.cardSerial.1 : 0
cardTable.cardTableEntry.cardSerial.2 : 0
cardTable.cardTableEntry.cardSerial.3 : 0
cardTable.cardTableEntry.cardHwVersion.1 : 1
cardTable.cardTableEntry.cardHwVersion.2 : 2
cardTable.cardTableEntry.cardHwVersion.3 : 2
cardTable.cardTableEntry.cardSwVersion.1 :
cardTable.cardTableEntry.cardSwVersion.2 :
cardTable.cardTableEntry.cardSwVersion.3 :
cardTable.cardTableEntry.cardSlotNumber.1 : 0
cardTable.cardTableEntry.cardSlotNumber.2 : 1
cardTable.cardTableEntry.cardSlotNumber.3 : 2
chassisSlots.0 : 3

```

Figure 65. Contents of the BROWSER.SAV File

3.3.2 Graphing MIB Variables through the Browser

To graph MIB values through the Browser, we selected the following two variables from the DECnet group:

- dnHellos - Provides the total number of hello messages received.
- dnReceived - Provides the total number of DECnet packets received.

These variables can be found in the following MIB tree:

```
* private
  enterprises
    cisco
      temporary
        decnet
          dnHello
          dnReceived
```

To graph dnReceived, you traverse the MIB tree as shown above until you can select the dnReceived MIB variable. Press the **Start Query** button and once you get a reply back in the MIB Values display box, you can then press the **Graph** button. A line graph window will appear, plotting this variable over time as shown in Figure 66 on page 66. Graphs for both dnHello and dnReceived are included.

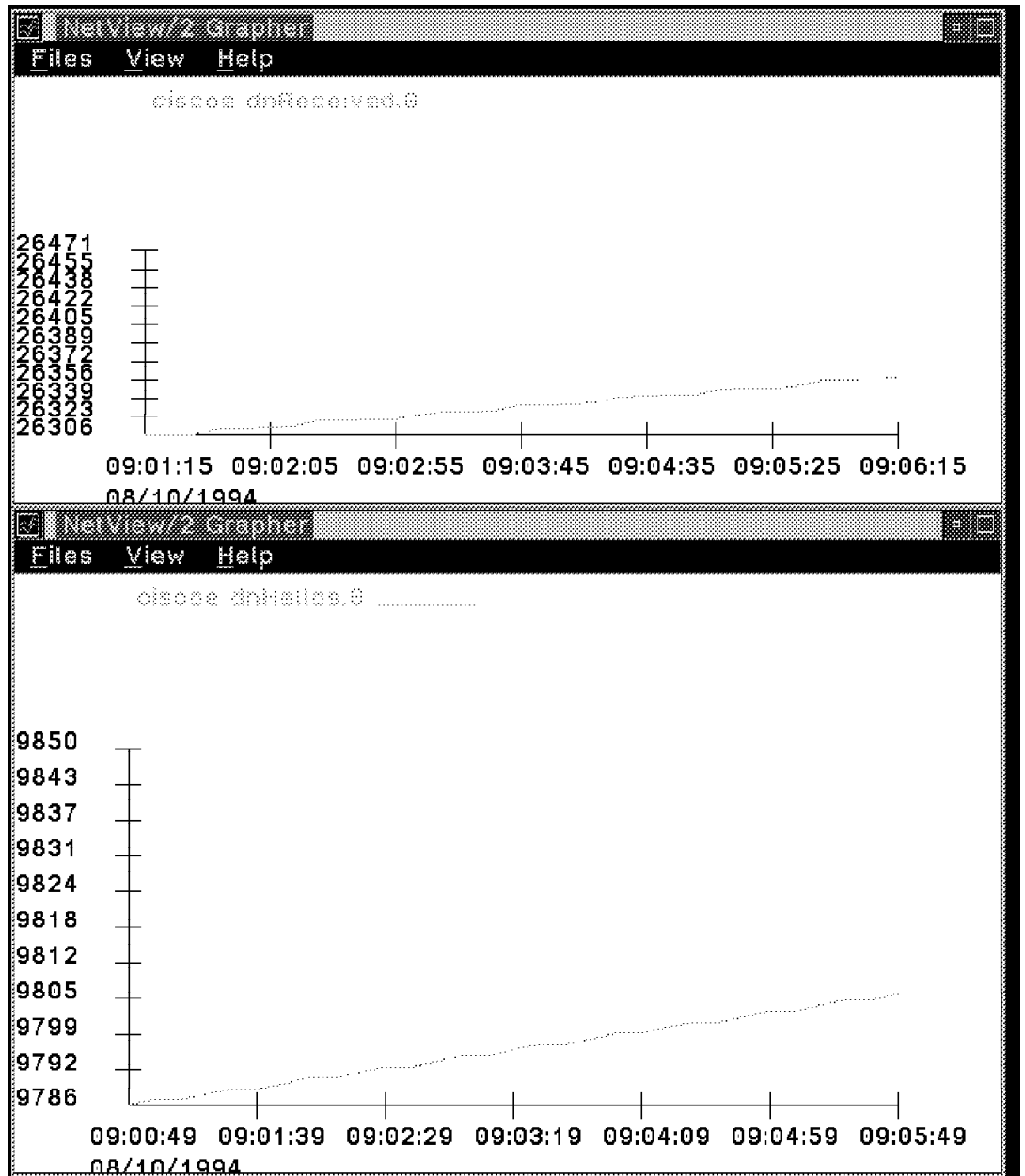


Figure 66. Example of MIB Variables Being Graphed (dnReceived and dnHellos)

3.4 Building and Executing MIB Applications

Certain MIB variables are accessed constantly and it is a tedious task to keep going down the tree using the MIB Browser, which can only return one MIB object at a time. The MIB Application Builder automates this process by providing a method of predefining the MIB tree path to a given set of variables. This generic query then can be used on any system by running this application directly from the MIB Application folder or by registering the applications, using the Registration application in the NetView for OS/2 folder. Then you can access the applications directly by clicking the right mouse button on one of the systems that is in a collection, and selecting **Application action** and the MIB application you wish to use.

This section will show the you how to create and execute some MIB applications that will provide you with some basic systems management reports. These reports are:

- Memory/Disk Storage Size/Used
- Graphing CPU Utilization

We will also show how to register these applications so that you can run them directly from your Management Desk. We will also show how you can run a selected application on a group of resources.

3.4.1 Memory/Disk Storage Size/Used

We will show the steps for creating a MIB application which returns the description, size and amount used for accessible disks on a system. We will then run this application on a sample system. The steps are as follows:

1. Open the IBM NetView for OS/2 folder on the desktop and double click on the Application Builder icon as shown in Figure 67.

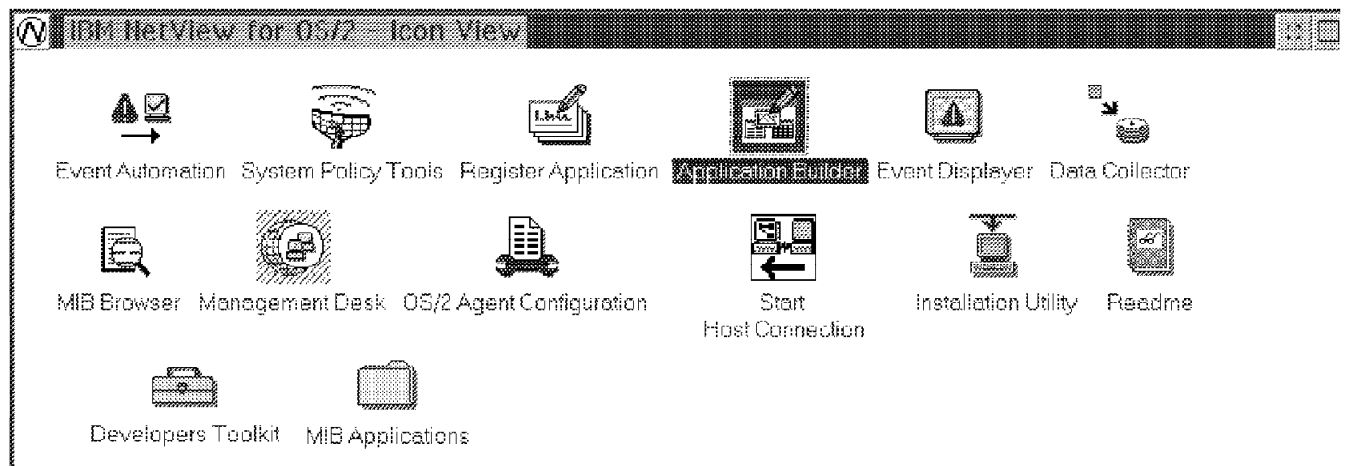


Figure 67. Icon for Application Builder in the Systems Folder

2. The MIB application window is shown in Figure 68 on page 68.

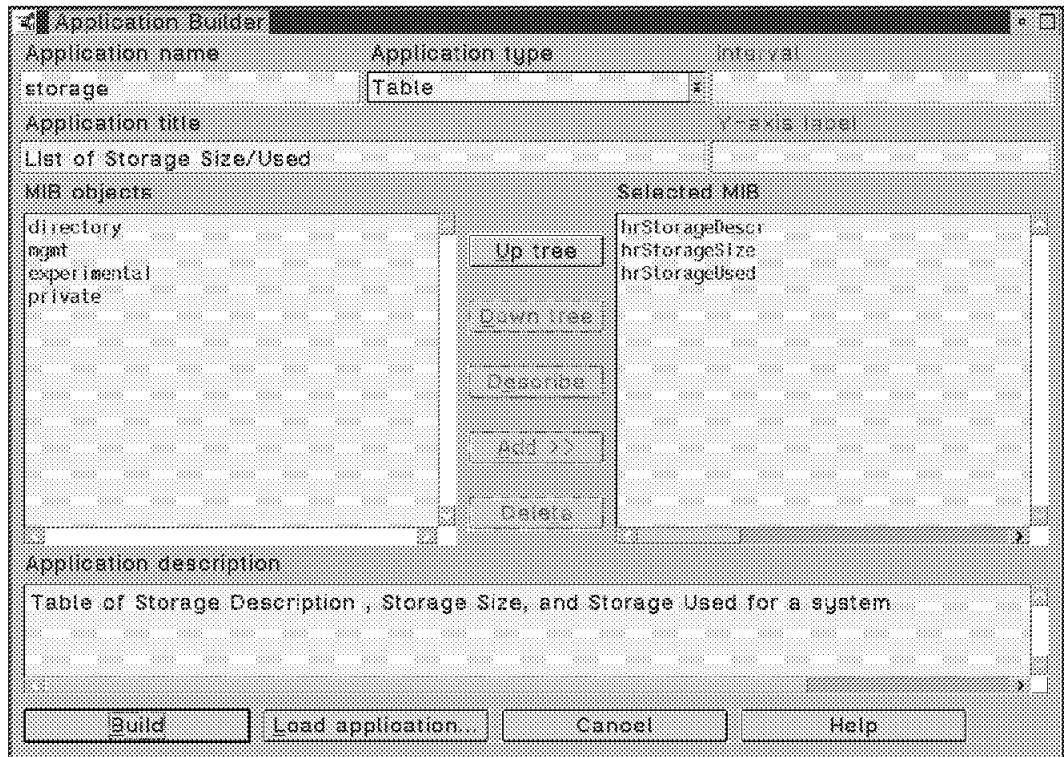


Figure 68. MIB Application Builder Screen

Fill in the fields as follows:

- a. Create a name for the application in the Application name field. This name is used to store all the entries that you put in this panel so that you can reference or change it later. All applications are stored in the \nv2\app\ directory as *Appname.app*. In this example, our disk storage application would be stored as *storage.app*.
- b. Expand the Application type field and click on table. The interval and Y-axis label fields are only used for graphs.
- c. Enter a title in the Application title field. This title is used to name the icon for this application in the MIB Applications folder and it is used by the Management Desk for its menu when it is registered to the Management Desk.
- d. Go down the tree in the MIB objects window to *hrStorageEntry* as follows:

```
*mgmt
  mib-2
    host
      hrStorage
        hrStorageTable
          hrStorageEntry
```


In the *hrStorageEntry* subtree highlight the following entries. Click on the **Add >>** button for each entry.

- hrStorageDescr
- hrStorageSize
- hrStorageUsed

When using tables you are allowed to enter objects from the same subtree. Attempting to enter tables from other subtrees will result in an error. You can add objects other than tables but these will not return a value when the query is executed.

- e. Type a description in the Application Description window.
3. Click on the **Build** button at the bottom of the window.
 4. If the build completes successfully an information window will appear as shown in Figure 69.

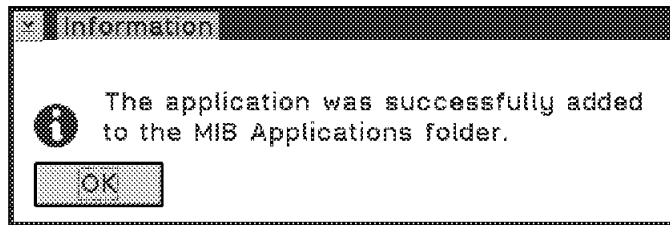


Figure 69. Successful Build of MIB Application

5. Close the Application Builder. You have now successfully built the MIB application. To run the application directly go to the MIB Application folder and double click on the **List of Storage Size/Used** icon. See Figure 70.

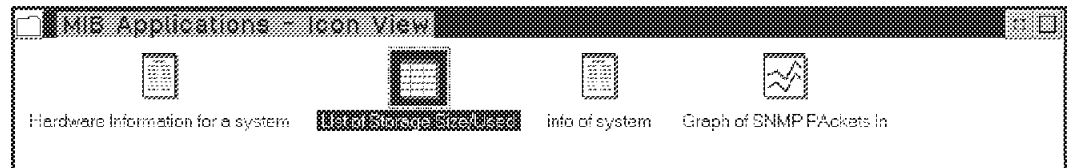


Figure 70. MIB Applications Folder

6. The Application Executor program will run this application and display a window for your new application. In the **Address** field enter the name of the machine for which you wish to retrieve information and click on the **Start query** button. The results of our query are shown in Figure 71 on page 70.

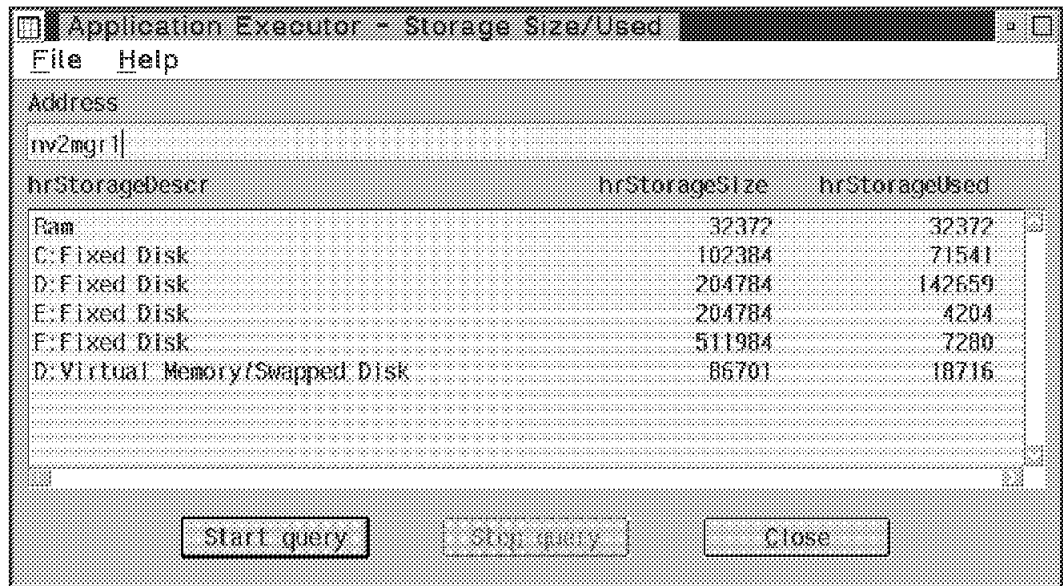


Figure 71. Sample Run of a Custom MIB Application

3.4.2 Graphing CPU Utilization

Before you can build a MIB application to graph CPU utilization busy, you must enable polling and data collection for this particular MIB variable.

Two methods can be used to enable a managed machine to be polled and allow the collection of CPU utilization data. After that the MIB application can be created for monitoring or graphing the CPU processor utilization:

1. Change the file C:\ANV2\ETC\CONFIG\SIA_BASE.CFG to include the following two lines:

```
siaProgramControlPerfInfo = 1
siaProcessorUtilizationPollingInterval = 20
```

where `siaProgramControlPerfInfo` is changed from 0 to 1 to turn data collection on and the `siaProcessorUtilizationPollingInterval` is changed from 0 to 20 seconds. These two lines can be added to the bottom of the `SIA_BASE.CFG` file.

OR

2. Using the MIB Browser for GET/SETs, here is the MIB structure to find these two elements and change them:
 - a. For `siaProgramControlPerfInfo`:

```

* private
  enterprises
    ibm
      ibmProd
        os2SIA
          siaProgramInformation
            siaProgramData
              siaProgramControl
                siaProgramControlPerfInfo

```

b. For siaProcessorUtilizationPollingInterval

```

* private
  enterprises
    ibm
      ibmProd
        os2SIA
          siaHostExtensions
            hrDeviceExt
              siaProcessorUtil
                siaProcessorUtilCommon
                  siaProcessorUtilizationPollingInterval

```

Now, that you have set up the variable you can build your CPU utilization MIB application.

1. Open the IBM NetView for OS/2 folder on the desktop and double click on the Application Builder icon.
2. Fill in the fields as follows:
 - a. Type in a name in the Application Name field. We used CpuUtil.
 - b. Expand the Application type field, and click on *Graph*.
 - c. Set the interval in the interval field to 20.
 - d. Enter % Utilization in the Y-axis label field.
 - e. Enter a title in the Application title field. We used CPU Utilization.
 - f. Go down the tree in the MIB objects window to siaProcessorUtilizationBusy as follows:

```

* private
  enterprises
    ibm
      ibmProd
        os2SIA
          siaHostExtensions
            hrDeviceExt
              siaProcessorUtil
                siaProcessorUtilCommon
                  siaProcessorUtilizationTable
                    siaProcessorUtilizationEntry
                      siaProcessorUtilizationBusy

```

Highlight *siaProcessorUtilizationBusy* and click on the **Add >>** button. The system will only allow you to graph MIB variables that are numeric. If you choose a MIB variable that is a character string, then the *Graph* will not even appear in the **Application Type** drop box.

g. Type a description in the Application Description window.

You should now have an Application Builder window just like what is shown in Figure 72.

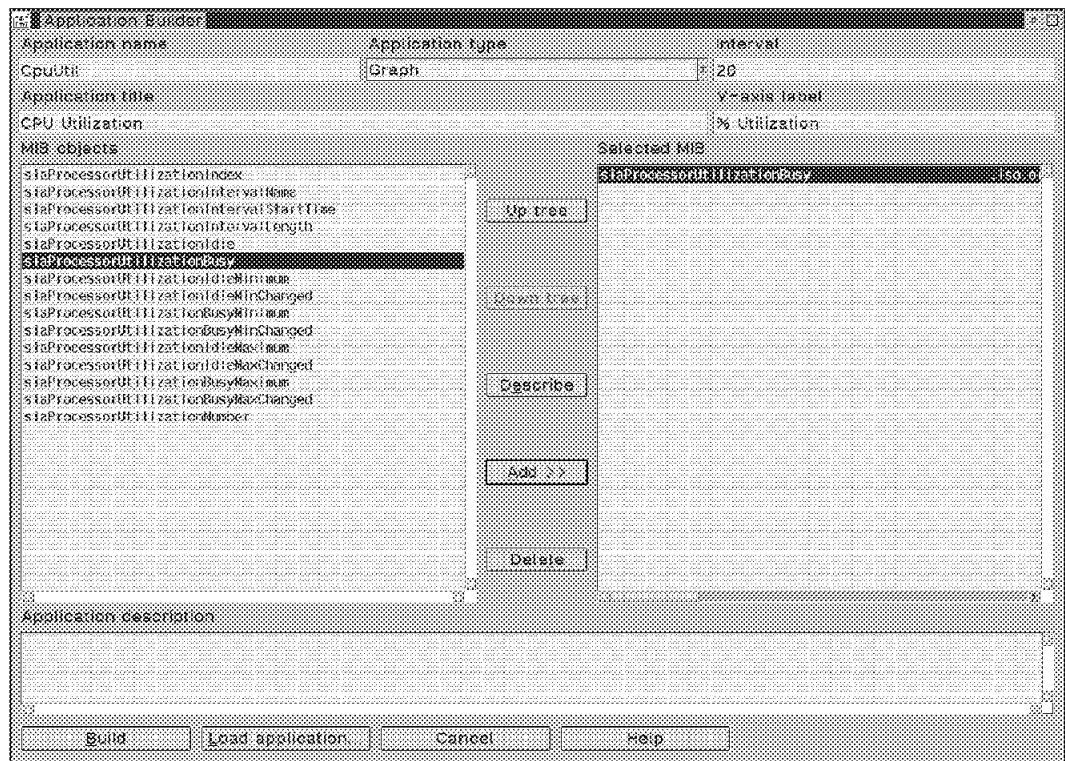


Figure 72. Application Builder Window for CPU Utilization Graph

3. Click on the **Build** button at the bottom of the window.
4. If the build completes successfully, an information window will appear.
5. Close the Application Builder by clicking on the **Cancel** button. You now have successfully built the MIB application. To run the application directly, go to the MIB Application folder and double click on the **CPU Utilization** icon

and you will get a graph that looks similar to what is shown in Figure 73 on page 73.

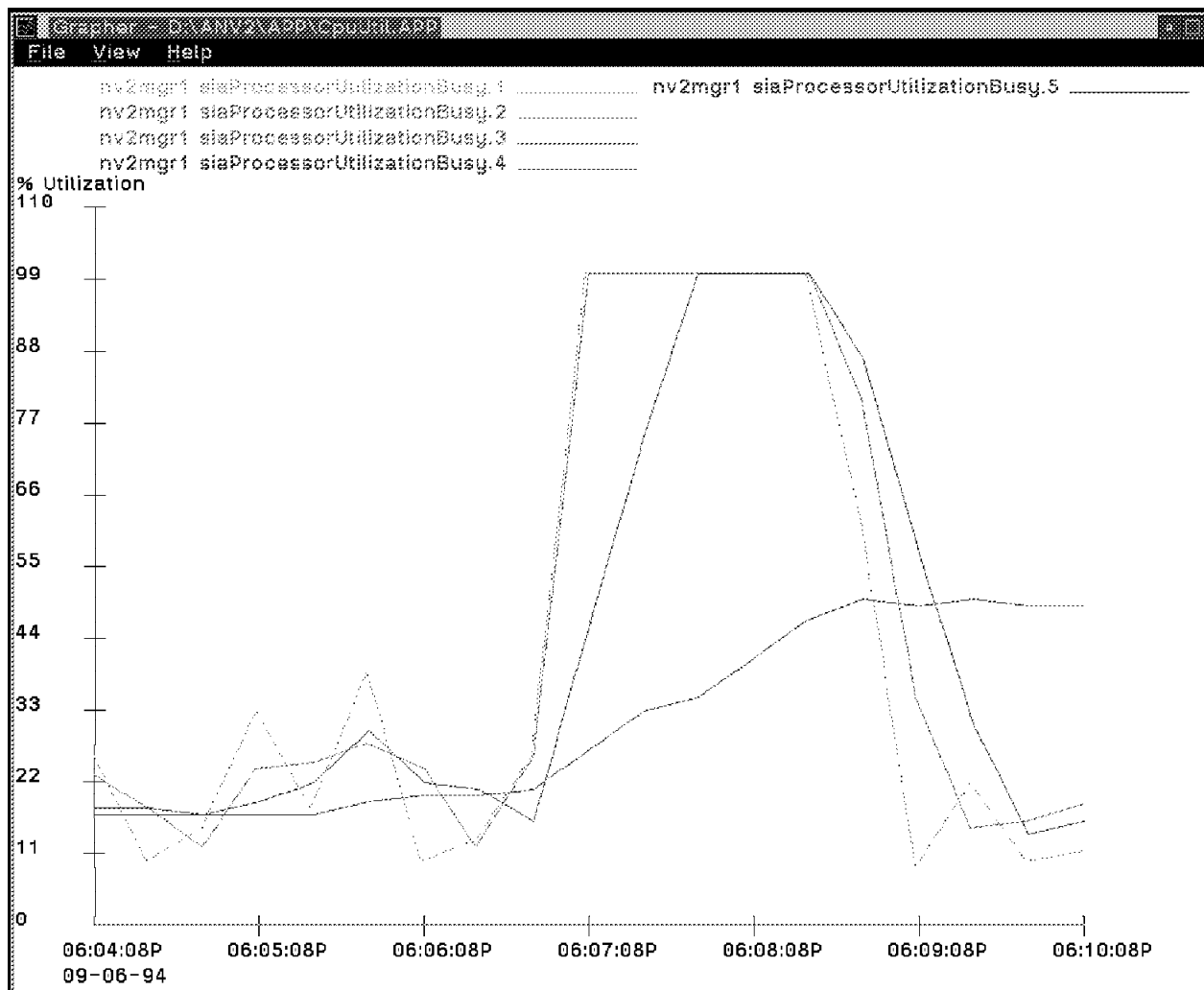


Figure 73. MIB Application Execution - CPU Utilization Graph

You will find a varying number of siaProcessorUtilizationBusy instances depending on how long you have been polling before graphing. As you can see, we had five instances of this MIB variable to graph. Following is a basic chart showing the average time used to calculate the processor utilization by instance number:

MIB INSTANCE	AVERAGE TIME
=====	=====
1	15 seconds
2	30 seconds
3	1 minute
4	5 minutes
5	15 minutes

3.4.3 Registering MIB Applications

Once you have created and saved your MIB applications, they are stored in the \anv2\app\ directory. This is the default directory. The storage application we created was placed in \anv\app\storage.app and looks like:

Table

```
Storage Size
3
hrStorageDescr 1.3.6.1.2.1.25.2.3.1.3
hrStorageSize 1.3.6.1.2.1.25.2.3.1.5
hrStorageUsed 1.3.6.1.2.1.25.2.3.1.6
Description/Size/Storage Used
```

To register the applications, we simply open the IBM NetView for OS/2 folder and double click on the **Register Applications** icon as shown in Figure 74.

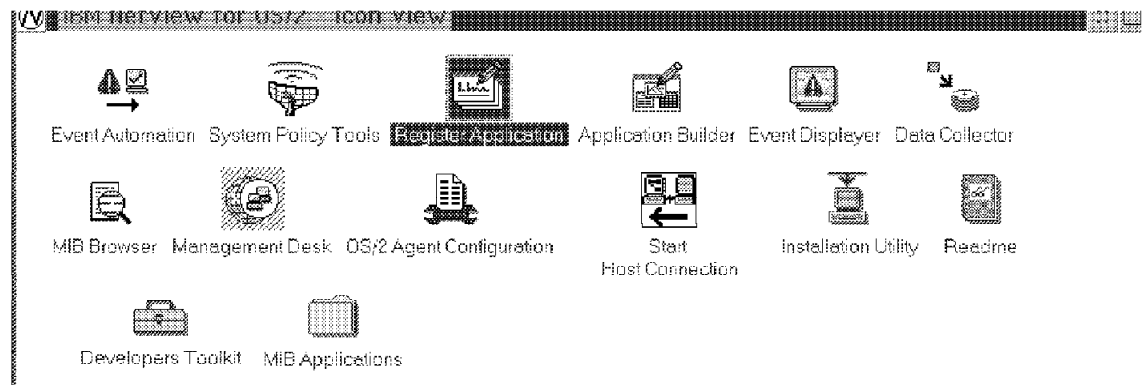


Figure 74. MIB Registration

An OS/2 window will open and all the MIB applications in the \anv2\app\ directory will be registered to the desktop. In this window, you should receive messages that your applications have been registered successfully.

3.4.4 Running Your Applications on a Group of Resources

After you have registered your MIB application you can access it from the Management Desktop directly by clicking on the right mouse button after you have highlighted a system or a group of systems. To run the *List of Storage Size/Used* application that you just created on a group of workstations, perform the following steps:

1. Select a group of systems by dragging the mouse across your management collection with the left mouse button pressed.
2. Click on the right mouse button and select **Application action...**
3. Click on **List of Storage Size/Used**.

Several windows will appear on the desktop (depending upon how many systems you selected). All of them will be running the storage application you selected.

3.5 Data Collector

With the Data Collector you can poll selected systems at regular intervals for MIB data. This data can be stored in a file, or used to generate events when the data exceeds a certain value and re-arm the thresholds when the data falls below a certain value. Should you decide to collect the data into a file, it will be stored in the \anv2\collect directory. The Data Collector can be used to keep track of important variables on certain systems, create an historical log of how these variables change over time, and initiate some automated action when the variable crosses a predefined threshold.

An example of this would be to track disk usage on certain machines and when a critical threshold is exceeded, generate events to begin the process to solve a problem. Once someone, or a task, is aware of a problem, problem resolution can begin. You may want to monitor CPU utilization for certain systems in your network and use this data to compare utilization of these machines over time. You can graph the results to get a better picture of what is happening in your network. You can determine which resources are being used the most in your network and use this data in management reports and to plan and control your network resources. Logging vital data can improve your view of the network by providing a history of network performance, utilization, peak times, down times and activities being done on the network over time.

Let us say for example that it is company policy to turn off machines at night. You want to keep track of the up time of certain machines to ensure that this policy is adhered to. We selected 2 machines which we will poll every 15 minutes to check the variable sysUpTime to see how long each machine has been running. The data collected on several systems can tell us the average up time of these machines, times rebooted during the day, whether a machine was running at a given time of day and other facts. The value of the data that is captured is usually not limited to just a snapshot of time. Its value often lies in its ability to help forecast future requirements and to understand trends. This historical progression can give greater insight into how the network is being utilized and where possible problems lie.

In order to set up and use the Data Collector, there are several steps that will have to be performed:

1. To start the Data Collector double click on the Data Collector icon in the IBM NetView for OS/2 folder. See Figure 75.

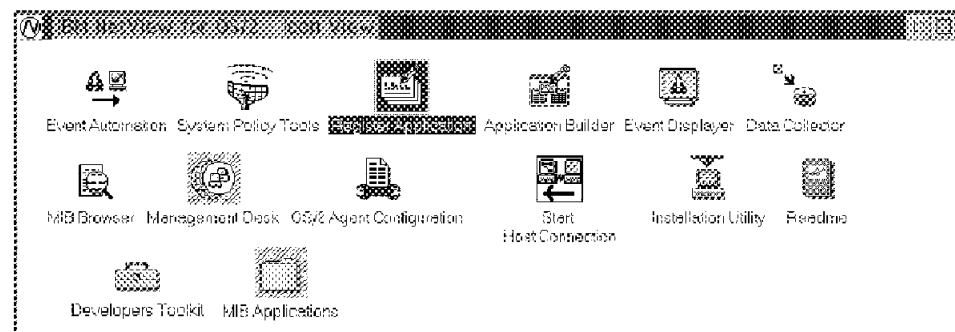


Figure 75. Data Collection Icon in NetView Folder

The main panel for the Data Collector will appear. The only active button the first time will be the **Add...** and all the fields are empty as shown in Figure 76 on page 76. If you have entered some values in an earlier session, they will show up in the MIB Object Collection Window. (If you have Deleted them earlier and they still appear, you probably forgot to press the **Apply** button after the Delete.)

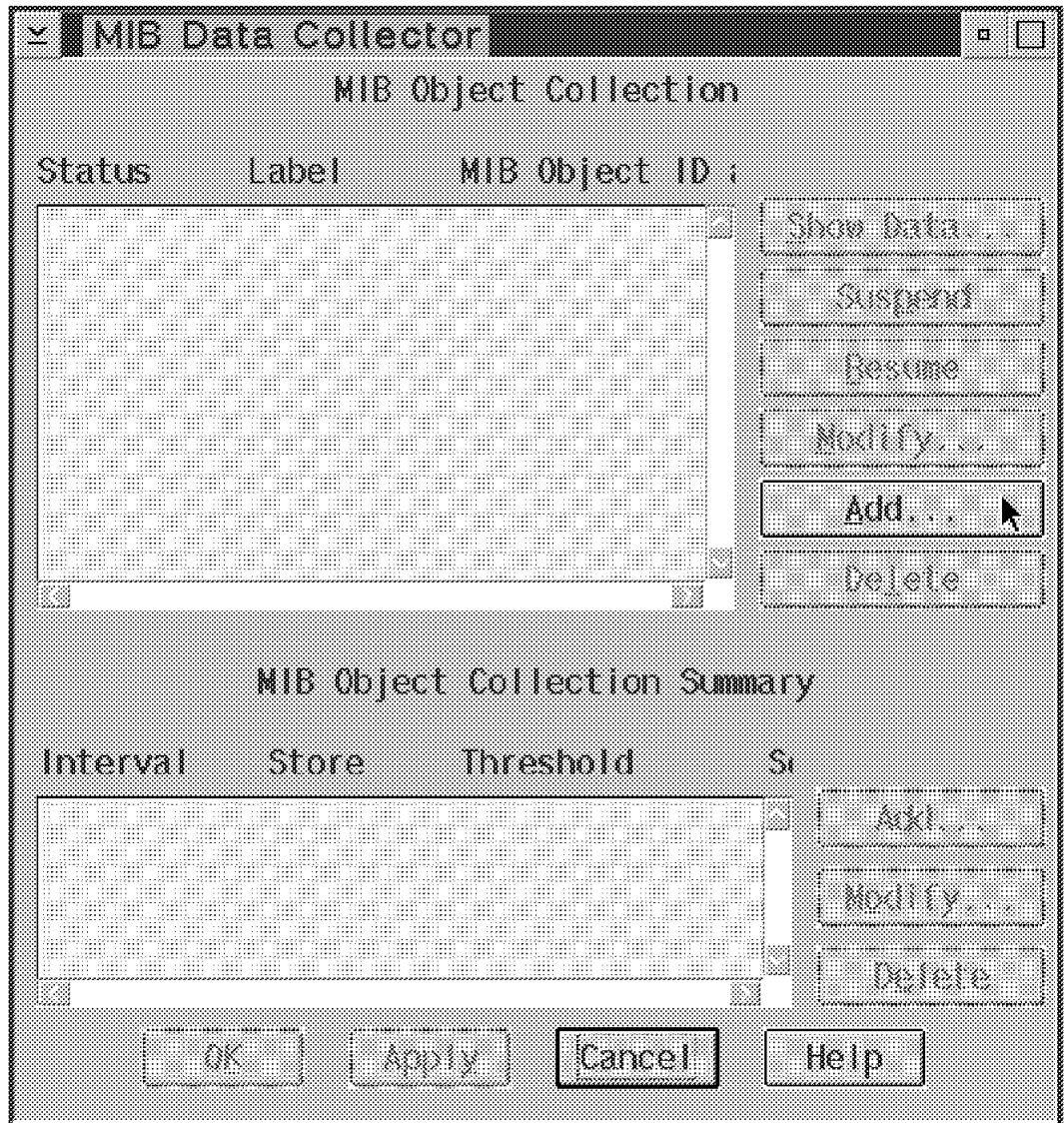


Figure 76. MIB Data Main Screen

2. Add a new object by pressing the **Add...** button. Another window appears with the MIB tree as shown in Figure 77 on page 77. We found that it is much easier to use the MIB Browser to find the variable first. The MIB Browser allows a query of the values which will be useful later if we have to set a threshold value. You will be able to collect only numerical data that is in one of the following formats:
 - a. Integer
 - b. Counter
 - c. Gauge

d. TimeTicks

You can always select the Describe button if you are not sure if the value is a valid one. A name will not appear in the *Label* box if it is not a valid value.

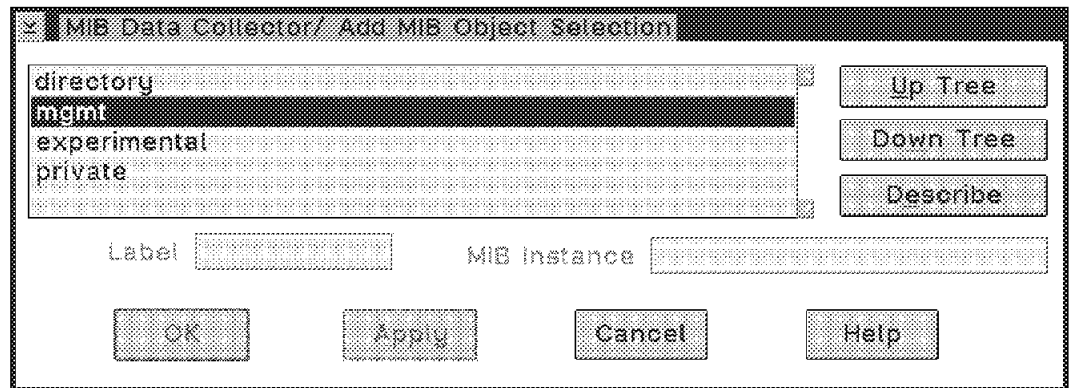


Figure 77. Selecting an Object in the Tree

Go down the MIB tree as follows:

```
* mgmt
  mib-2
    system
      sysUpTime
```

3. Select SysUpTime. Go to the field labeled File and fill in a file name. The label is used as the file name under which the data will be stored in the `lanv2\collect` directory. The MIB instance field contains the number of the instance you wish to keep track of or a `.*` to indicate all instances are to be stored. The naming convention used for the files is `label.instance`, where `label` is the name you enter in the label field and `instance` is an integer between 0 and the maximum number of instances for the selected MIB object. If for example we want to use a label named "myname" and want to select all instances of our object (for example, 3 instances), enter `myname` in the **Label** box and a `.*` in the **MIB Instance** box, and the following files will be created:

```
myname.1
myname.2
myname.3
```

If there is only one instance and you do not enter a number, a 0 is used. We entered the Label as `OnSystem` and a MIB instance of 0 since only one value exists for this variable. Click on the OK button. See Figure 78 on page 78.

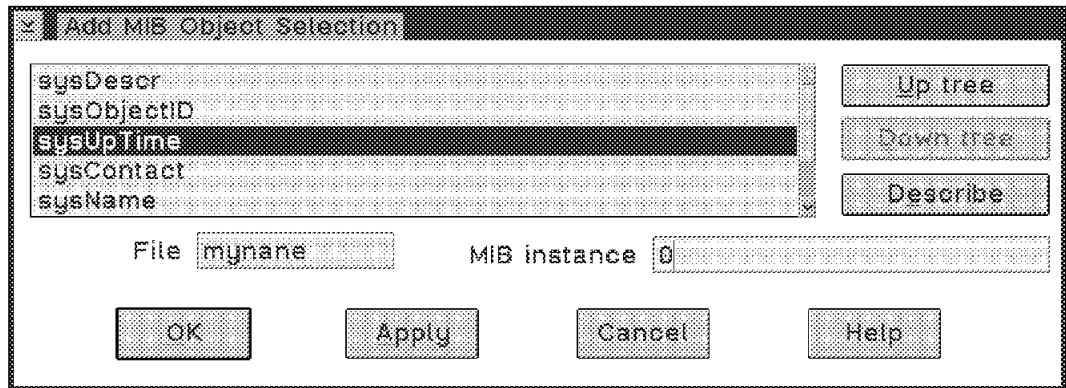


Figure 78. Entering a Label Name

4. The next screen defines for which systems you will be collecting data. See Figure 79 on page 79. The Host Name or Address is the system you wish include in the collection process. You can select one of 3 choices for collection mode. Each has a different impact on how data is processed from the polled system.

- a. *Store, Check Threshold.* - Use this to keep a log of all the collected values and generate an action when defined thresholds are exceeded. When collecting data for a short polling interval or over a long period of time the files in which the data is stored can grow very large. Make certain that you have enough space to store all the data and clean your \anv2\collect directory of unwanted files periodically.
- b. *Store, No Thresholds* - If you only want a log of collected data for future reference or analysis without any special actions, select this option. Be careful of how you define the polling interval or for how long you are collecting data because the files can grow very large when left unattended.
- c. *Don't Store, Check Thresholds.* - Select this one if it is not important to store values, but still allow threshold monitoring.

Caution

Remember that data collection is active even when the Data Collector is closed. The NV2KC.exe continues to store values as long as the Object is not Suspended. See page 80 on how to suspend collection.

5. Select the polling interval using s, h, d, or w after the number to denote seconds, hours, days or weeks. We typed in 15m.

If you selected *Store, No Thresholds* click on the OK button. You do not have to fill in the rest of the values.

6. Trap number 58720263 is the SNMP trap number which will be sent to the managing system when a trap is triggered. When a trap is sent, it appears in the managing system event log. You can examine these traps by using the Event Displayer located in the NetView for OS/2 folder.

MIB Data Collector/ Add Summaries for myname

Host Name or Address: nvcclient

Collection Mode: Don't Store, Check Thresholds

Polling Interval: 30s Trap Number: 58720263

Threshold >: 50 Rearm <=: 75 ☒ Percent ☐ Absolute

Threshold Action:

Rearm Action:

OK Cancel Help

Figure 79. Setting the Thresholds and Storing Data

Thresholds are used when you want to check if the value you have selected has been exceeded, and you wish some action to be taken. In our case we would like to set a threshold of 10 hours for the MIB variable SysUpTime. The system will respond by sending a trap to the managing station when this value is exceeded to inform us that the machine has been running more than a full workday. The SysUpTime is measured in ticks which represent 1/100 of a second. The threshold value would then be 3,600,000 ticks (100 * 60 sec * 60 min * 10 hrs). The Data Collector polls the host system and compares the value it gets with this threshold value. If it has exceeded the threshold, then the action specified in the **Threshold Action** field (see Figure 79) is executed. To prevent execution of the command every time the station is polled, a rearm value is used. This value can be a percentage of the threshold value or an absolute number relative to the threshold value. In our case, since the variable is growing constantly, the rearm value can be set to anything below 100 percent or below 3,600,000 ticks.

7. We left the **Threshold Action** and **Rearm Action** fields blank since we only want a trap to be sent which will be processed by Event Automation. The Threshold Action and Rearm action can be any OS/2 command including a command file.
8. Click on the **OK** button to continue.
9. Click on **Cancel** in the Add MIB Object Selection window shown in Figure 77 on page 77 to return to the main window shown in Figure 80 on page 80.

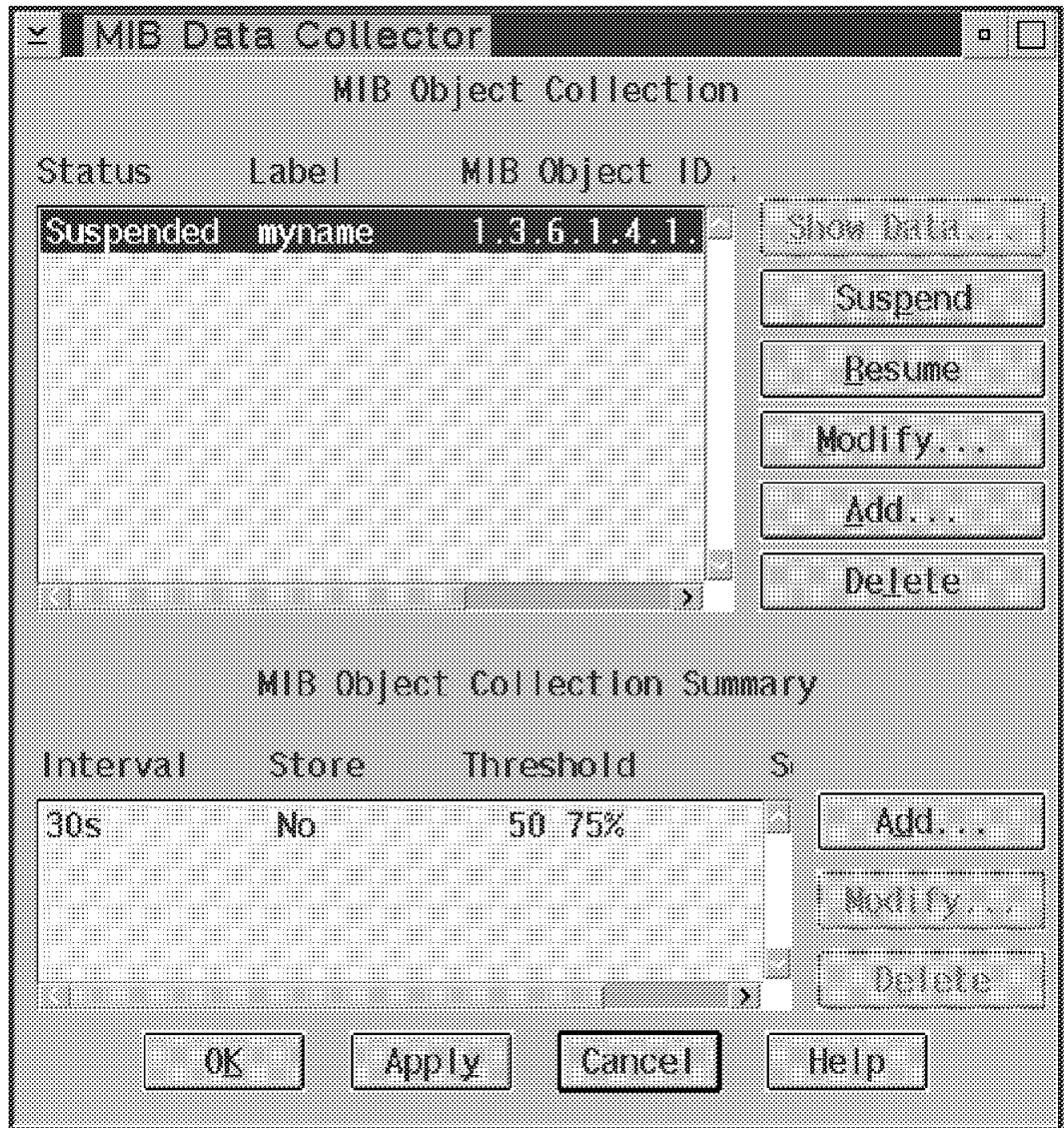


Figure 80. Data Collector Window

Once you have returned to the main window you will see your newly created object in the MIB Object Collection window. The status will be suspended. This means that no data is being collected. This is the only way to stop collection of data.

10. Start the data collection by selecting your object and clicking on the **Resume** button. Data will be collected even if you close the MIB Data Collector.

Note: You will have to wait for one collection interval before the **Show Data...** button is active.

11. We have defined only one system for which we will be collecting data. We can add a second system by clicking on the **Add** button and entering the same data as shown in Figure 79 on page 79 and changing the host name or address to 9.24.104.42. This TCP/IP address belongs to an XStation. Since it has a MIB with the SysUpTime variable, we can monitor it as well. Click on the **OK** button. You should now have two entries in the MIB Object Collection Summary.

12. Click on **Apply** in the MIB Data Collector Window. We started all our machines in the morning and left collection running for the day. At the end of the day we clicked on the **Show Data...** button. A window showing the collected data appears as shown in Figure 81 on page 81.

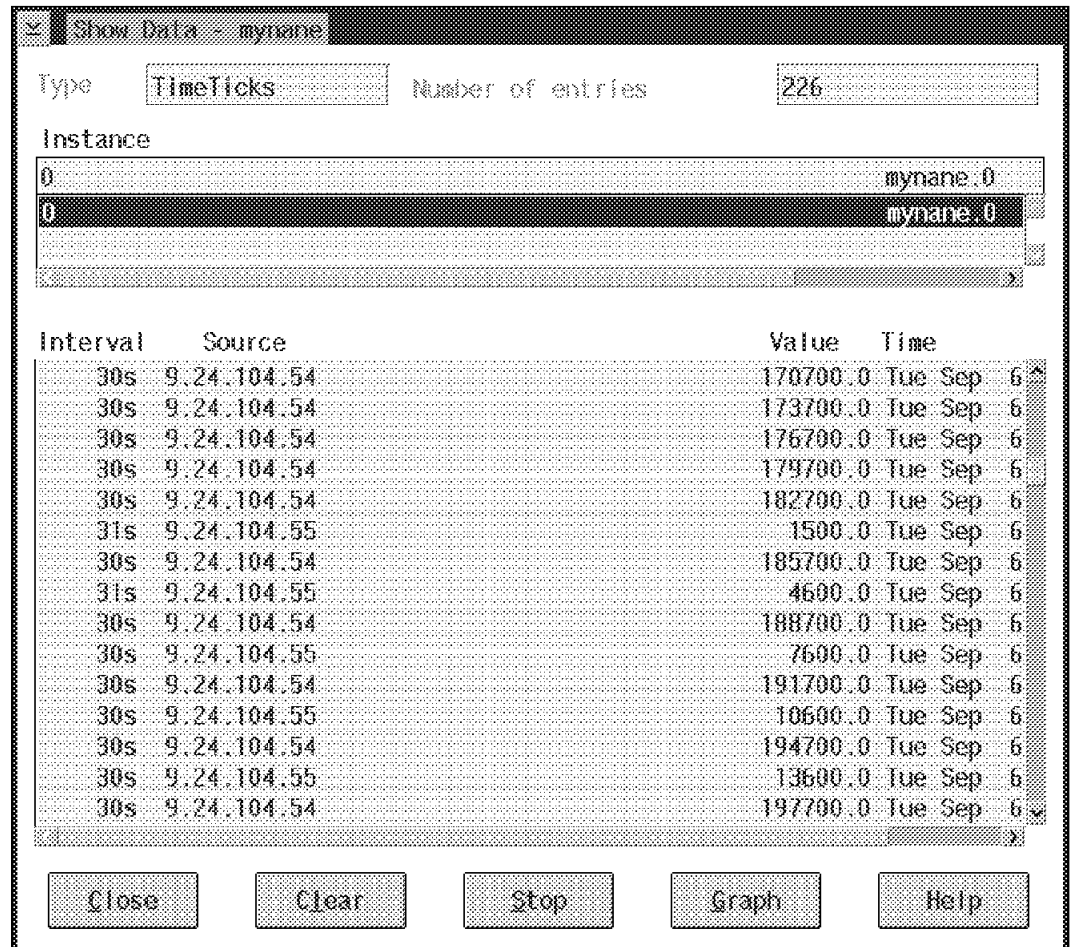


Figure 81. Show Data Collected

In Figure 81 you see all of the data collected for the two machines during the day at 15-minute intervals. You can scroll up and down the list to see all the values. Click on the **Graph** option and a window with a graph of the results will appear.

An example of what the graph would look like is shown in Figure 82 on page 82.

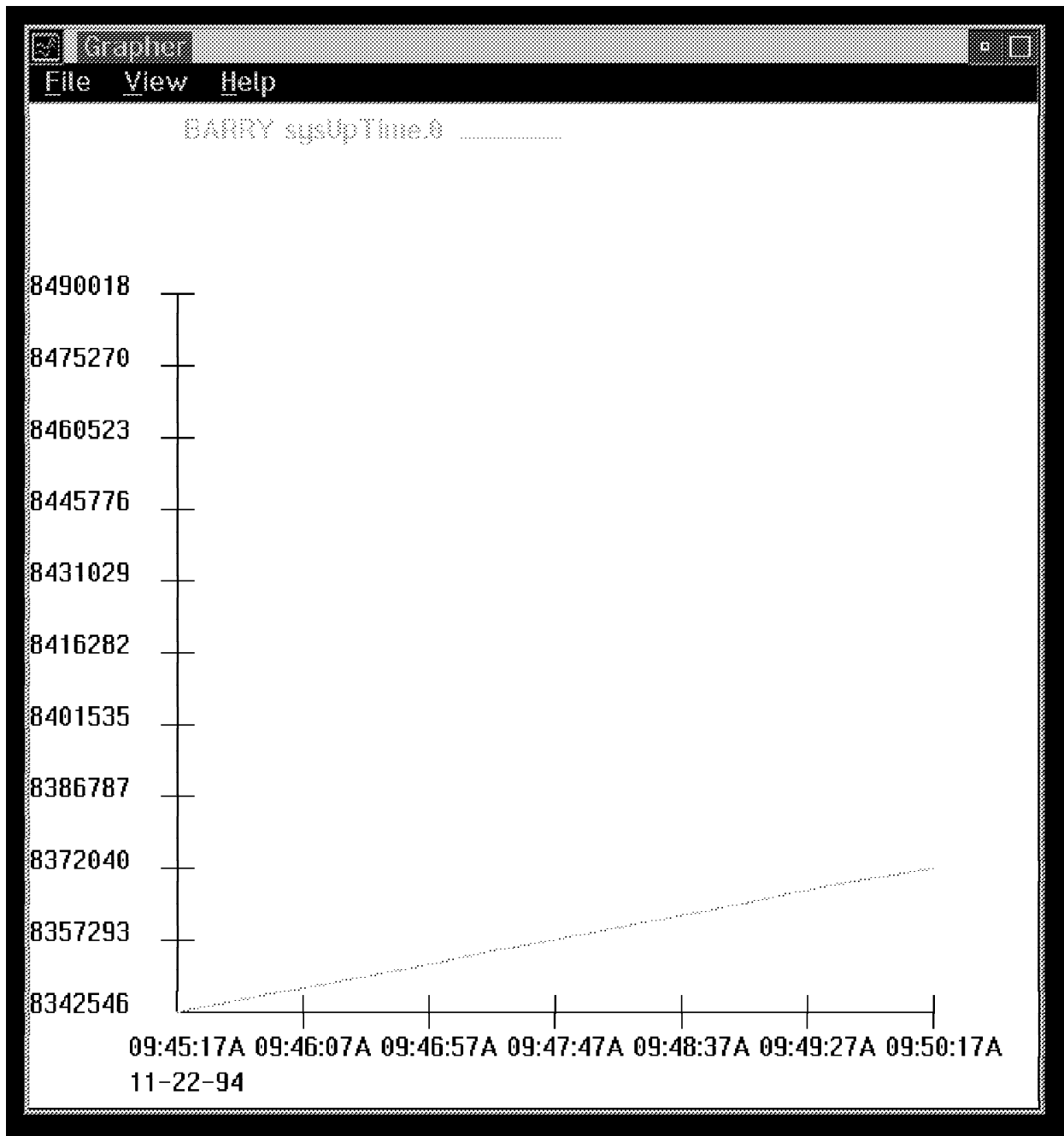


Figure 82. Show Data Graphed

To get a more readable graph do the following:

- Select View from the pull-down menu and click on **time intervals**. Enter a value of 10h.
- Select View from the pull-down menu and click on **show counter as...** and click on **Actual sample value**.

We can use the graph or the data to gain additional information about:

- When the machine was turned on (ticks are a small number)
- How long and when it was on (time is saved with collected value)
- When it was turned off (no data collected for that period)

- d. How often it was on and off over a certain period of time.
13. The approximate schedule of when the system was running can be helpful when dealing with security matters (for example, the room or building in which the computer was located was supposed to be empty) or scheduling access times for certain systems.

A well thought out collection strategy can bring a lot of useful information. We selected a very simple case to illustrate just how such collected data can be used. One variable over a period of time can tell us a great deal about what is happening in our network.

3.6 Event Automation and the Event Displayer

Throughout this book, you will see many examples and scenarios of how the Event Automator and the Event Displayer are used to trap and display events generated by managed systems. In this section, we will give an overview of how to set up the Event Automator to trap on operational state changes from an OS/2 LAN Requester. This will include showing how to see the alert on the Event Displayer and showing the system-supplied pop-up message as an automated action.

3.6.1 Setting Up the Event Automator

Before using the Event Automation application, you must ensure that both TRAPD and the Event Automation daemon are running on the managing system. You must also have loaded the required MIBs that you want to monitor. By installing the Agent code on your managing station, you should have all the required SNMPMIB data that you need.

To get started, double click on the **Event Automation** icon in the NetView for OS/2 Main Icon View window as shown in Figure 83.

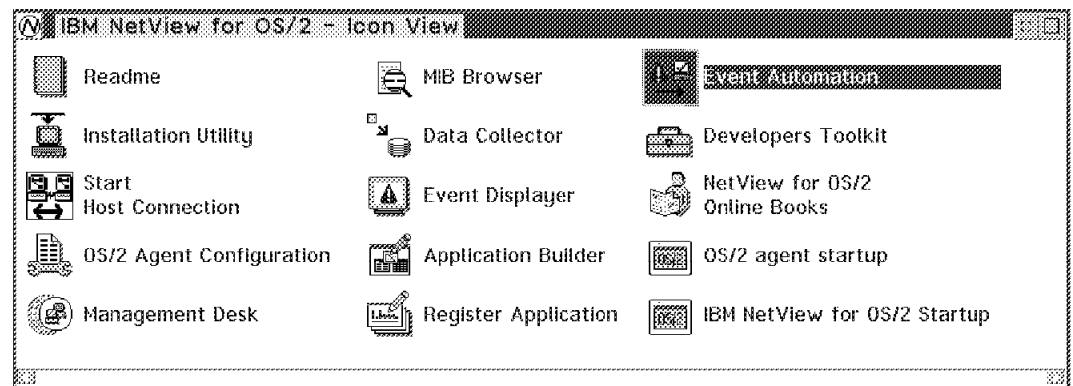


Figure 83. NetView for OS/2 Main Icon View

This will take you to the *Event Automation Update* main window. The very first time you access this window, all the fields and list boxes will come up completely empty as shown in Figure 84 on page 84.

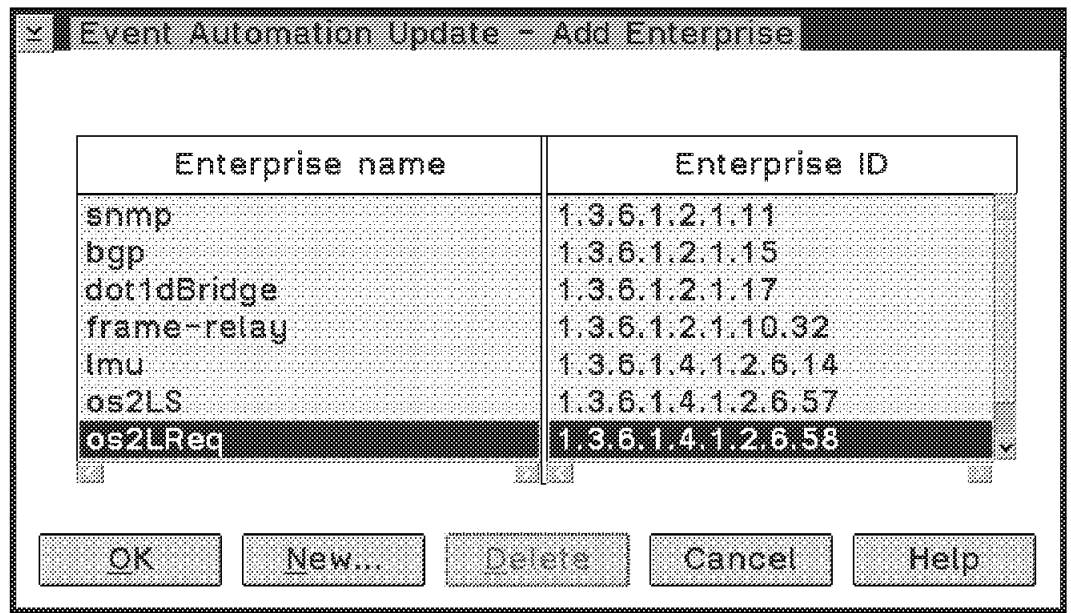


Figure 85. Event Automation - Add Enterprise Name and ID

2. Since we want to be monitoring for OS/2 LAN Requester events, click on the **os2LReq** line item in the *Enterprise name* list box and then click on the **OK** button. This will add the Enterprise Name and ID to the *Enterprise name and ID* list boxes as shown in Figure 86.

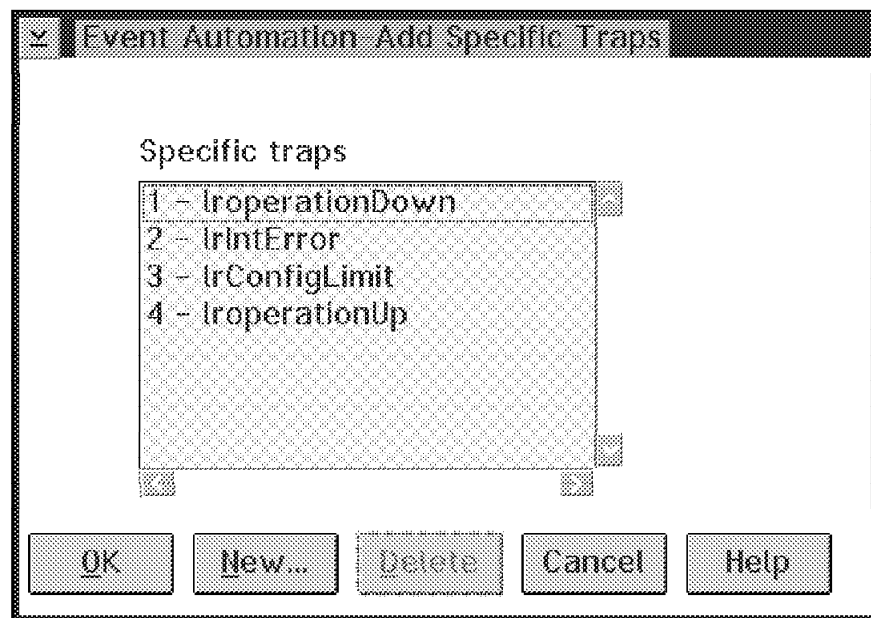


Figure 86. Event Automation - LAN Requester Trap Name and ID Added

3. We must now specify which Generic and Specific Trap types we want to capture. To do this, we must click on the **os2LReq** line item that we just added to the *Enterprise name* list box and then click on the **Add...** button under the *Generic Trap* list box. This will present the *Event Automation - Add Specific Traps* window as shown in Figure 87 on page 86.

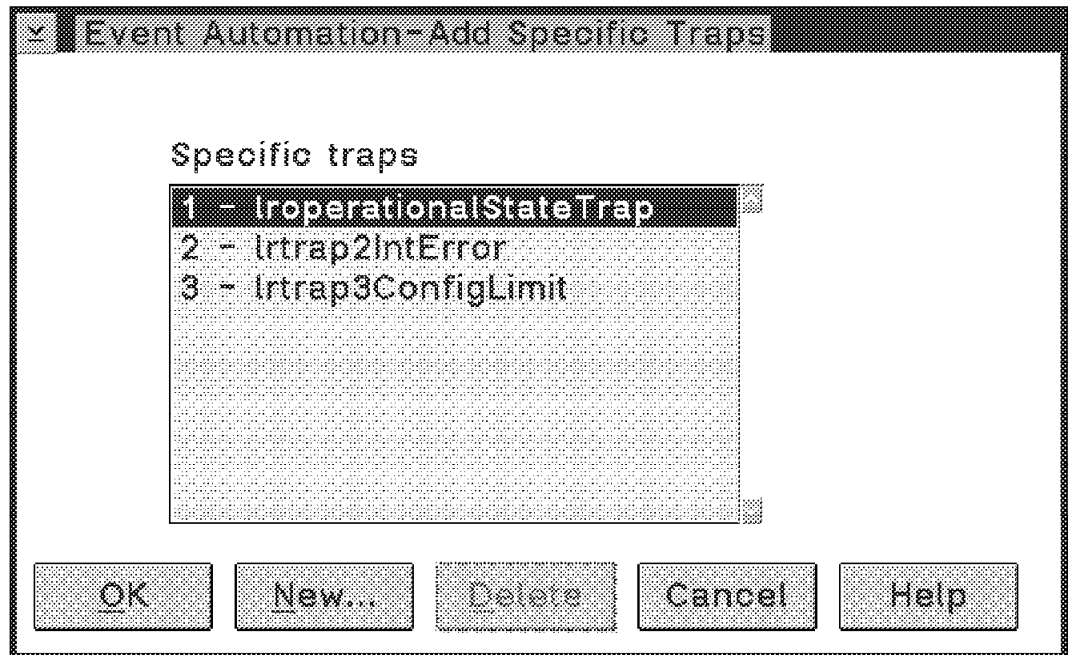


Figure 87. Event Automation - Adding Specific Trap Information

4. We must now locate the *Operational State Change* trap. It is the first trap type in the list box. Therefore, just click on the **1 - IroperationalStateDown** line item and then click on the **OK** button. This will add our LAN Requester Specific Trap type to the 2 rightmost list boxes in the *Event identification* area of the window as shown in Figure 88 on page 87.

Event Automation Update

Event identification

Enterprise name	Enterprise ID	Generic trap	Specific trap
os2l Reg	1 3 6 1 4 1 2 6 58	6 - enterpriseS	1 - Iroperation

Add... Delete Add... Delete

Action specification

☒ Popup message

Alias Message

☒ Pager

Address

☒ Forwarding

New status

☒ Status

Optional command(s): Find...

OK Apply Reset Cancel Help

Figure 88. Adding Specific Traps and Selecting Actions for These Traps

5. We must now select the Generic/Specific trap that we just entered. Ensure that both the *Enterprise Name/ID* and the *Generic/Specific trap* list boxes have our LAN Requester line item selected (highlighted) as shown in Figure 88. This will allow all the check boxes in the *Action Specification* area to become active.
6. Click on the *Pop-up message* check box so that we get the system-supplied pop-up message when the alert comes in. This is shown in Figure 89 on page 88.

Note that you can perform a number of actions for each alert that comes in. In Figure 89 on page 88, we also show how you can change the *Status* of a particular resource to **Degraded**. In our example, we will de-select the *Status* check box, and just have the Pop-up message be presented when the alert arrives.

Event Automation Update

Event identification

Enterprise name	Enterprise ID	Generic trap	Specific trap
os2lReq	1.3.6.1.4.1.2.6.58	6 - enterpriseS	1 - lroperation

Add... Delete Add... Delete

Action specification

☒ Popup message

☐ Pager Alias Message

☐ Forwarding Address

☒ Status New status

Optional comma Degraded Failed Degraded OffLine Unknown Find...

OK Apply Reset Cancel Help

Figure 89. Event Automation - Adding Automation Actions

- Once all of our actions have been entered into the preceding window, we can click on the **OK** button to save our updates and close the window.

Once we close this window, the Event Automation application will store our event-action record in the \anv2\etc\EVENTACT.CFG event-action file. This is a binary file that the Event Automation program uses when waiting for and processing events. When the LAN Requester Operational State Change trap is detected, the data from this event MIB is compared to what we entered into the event-action file, and since there is a match, our associated action (displaying a pop-up message) will be taken.

3.6.2 Using the Event Displayer

Now that we have set up the Event Automator to trap on our LAN Requester alert, we will show how to use the Event Displayer to view that event once the alert has been generated. To access the Event Displayer from the NetView for OS/2 Main Icon View, just double click on the **Event Displayer** icon as shown in Figure 90 on page 89.

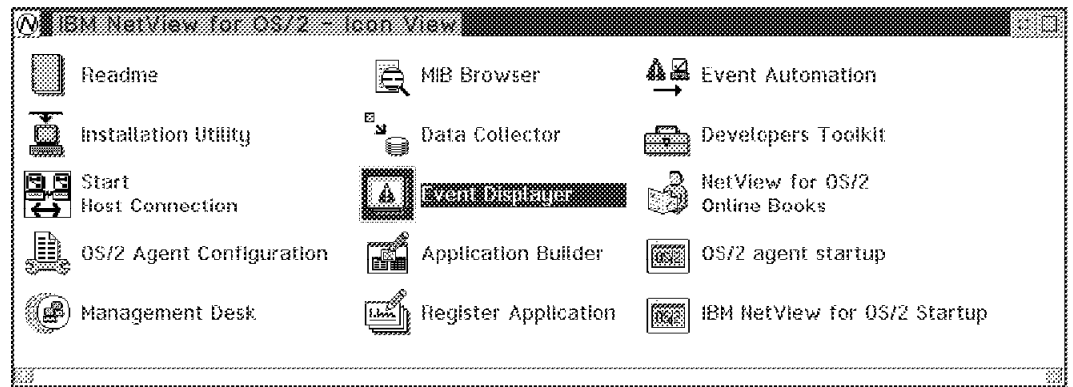


Figure 90. Selecting Event Dispatcher from the NetView for OS/2 Icon View

We will be presented with the *Event Dispatcher* window as shown in Figure 91. This window gives the user a variety of methods to limit the number of event records displayed. The defaults are set to display everything in the event log and the trap log. Before continuing with our scenario, we will briefly explain the types of events and traps that are possible, where they are recorded, and what all the display filters are.

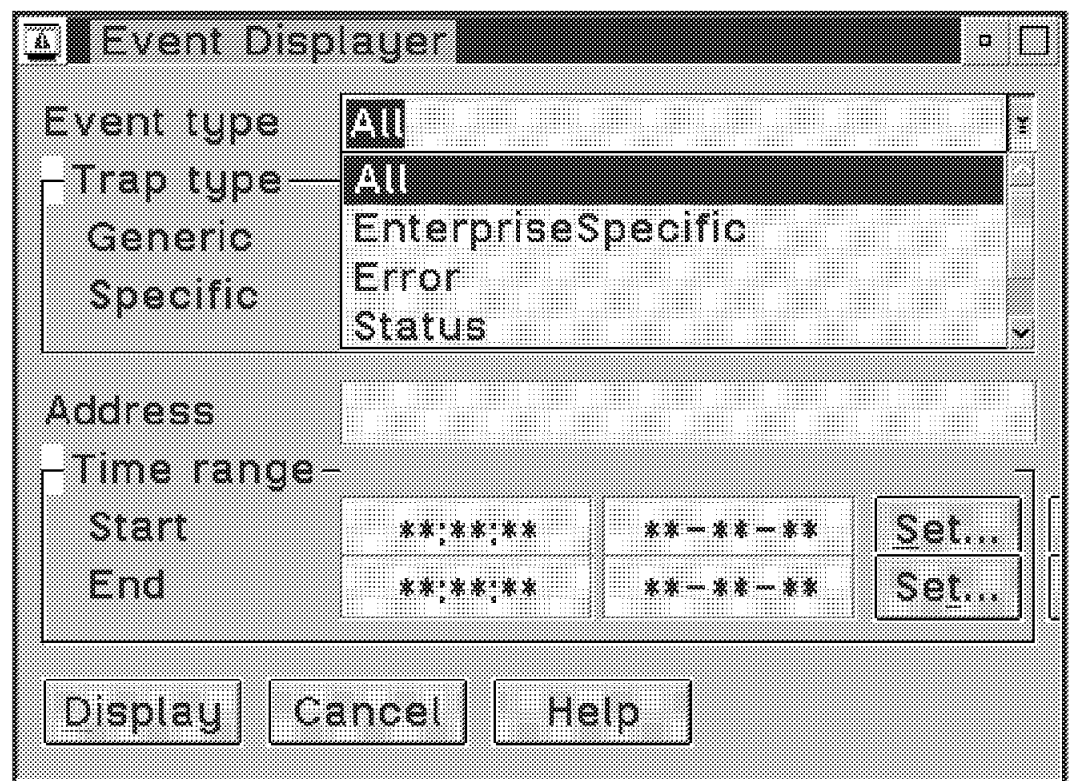


Figure 91. Event Dispatcher Window Showing Event Types

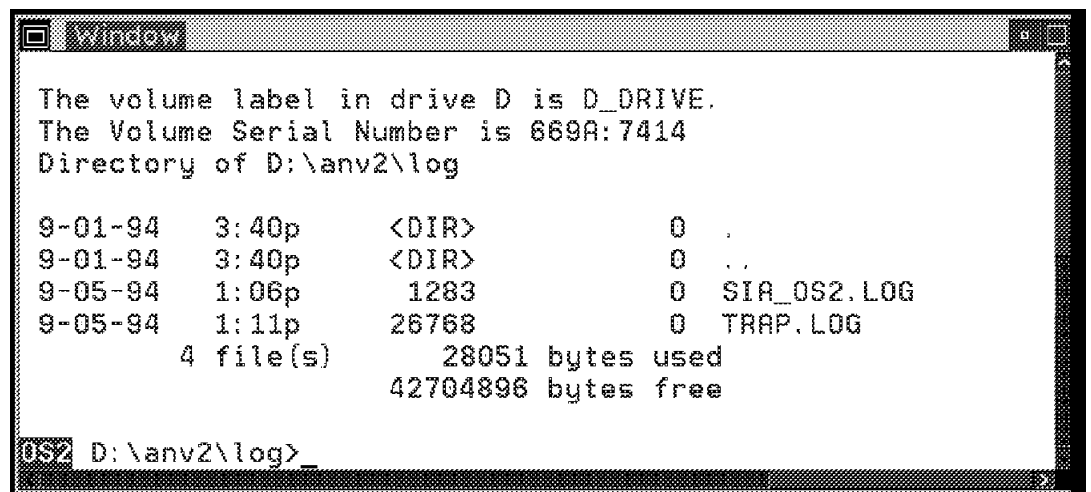
3.6.2.1 Event Types

From the preceding window, we can see that there are three different event types. The following table gives SNMP trap numbers and descriptions for each event type.

Table 1. Possible Event Types and Descriptions		
Event Type	SNMP Trap Number	Description
STATUS	0	Cold Start
	1	Warm Start
	2	Link Down
	3	Link Up
ERROR	4	Authentication Failure
	5	EGP Neighbor Loss
ENTERPRISE	6	Enterprise Specific

The above event types can also be divided into 2 categories. The 2 categories are defined by whether or not an event occurred as a result of some threshold being exceeded or rearmed. Only those events that report a threshold being exceeded or rearmed are stored in the EVENT.LOG. All others are stored in the TRAP.LOG.

The window shown in Figure 92 shows that these logs are stored in the \anv2\log directory. Note that since no *Threshold Exceeded/Rearmed* events have occurred, no EVENT.LOG file exists. NetView for OS/2 will create one as soon as one occurs.



```
Window
The volume label in drive D is D_DRIVE.
The Volume Serial Number is 669A:7414
Directory of D:\anv2\log

9-01-94   3:40p   <DIR>           0  .
9-01-94   3:40p   <DIR>           0  ..
9-05-94   1:06p   1283            0  SIA_OS2.LOG
9-05-94   1:11p   26768           0  TRAP.LOG
          4 file(s)      28051 bytes used
                               42704896 bytes free

OS/2 D:\anv2\log>
```

Figure 92. OS/2 Window Showing Location and Size of Logs

Trap Log Warning

It is the responsibility of each site managing station to monitor the size of this TRAP.LOG. It is recommended that it be printed and cleared periodically so that it never gets full. The maximum size of the trap log is 180KB.

Besides filtering on event type you can also limit the display of traps in the following ways:

- SNMP generic trap type/number as shown in Figure 93.
- Specific trap number for your enterprise-specific trap. For example, if you are using the Data Collector to monitor CPU Utilization Busy and send Trap number 1000 when your threshold of 80% is exceeded, then you could enter **1000** in the *Specific Trap Type* field, and the Event Displayer would show you only your CPU Utilization Exceeded events.
- By address or name of monitored resource. For example, if we only wanted the traps from our NetView client machine, then we would enter **nvclient** (the IP hostname) into the *Address* field.
- By the time range the event may have occurred.

We knew that our LAN Requester alert was an enterprise-specific trap, so we chose to display only those events that have a trap type of **6-enterpriseSpecific** as shown in Figure 93.

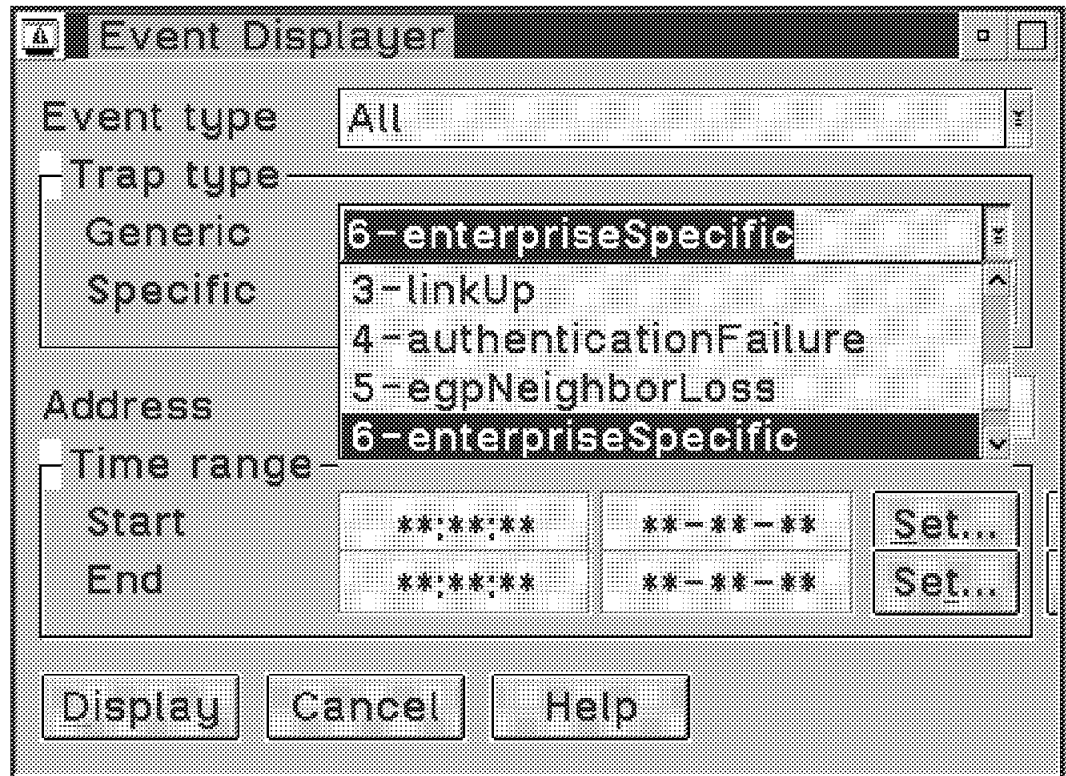


Figure 93. Event Displayer - Selecting Only Enterprise-Specific Traps

We went over to our NetView client machine which has an IP address of **9.24.104.68** and caused the LAN Requester to go down. This caused the alert to be sent to the managing station. If we click on the **Display** on the preceding Event Displayer window, we will get the *All Events* window as shown in Figure 94 on page 92. Note the highlighted alert has in fact come in from **9.24.104.68** and that it has a Generic alert type of 6, meaning that it is Enterprise Specific. Note also in the *Description* that the **IrOperationState=0**. This means that the LAN Requester has gone down. Please check the following event in the *All Events* window. When we bring the LAN Requester back up, we see that the **IrOperationState=3**. This means that the LAN Requester is up and running fine.

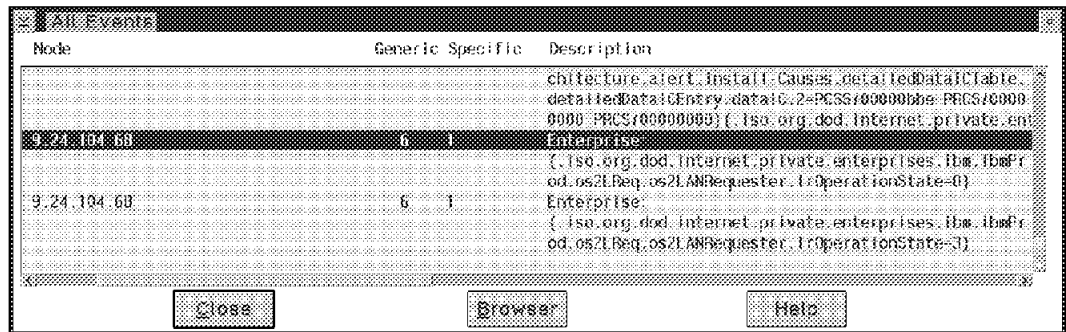


Figure 94. Event Displayer - All Events Window

Now to tie in to the Event Automator, remember that we included one action to be performed when the LAN Requester went down. This action was to display the system-supplied message pop-up about an operation state change. As soon as we took the Requester down, the pop-up shown in Figure 95 was displayed on the NetView for OS/2 managing station console.

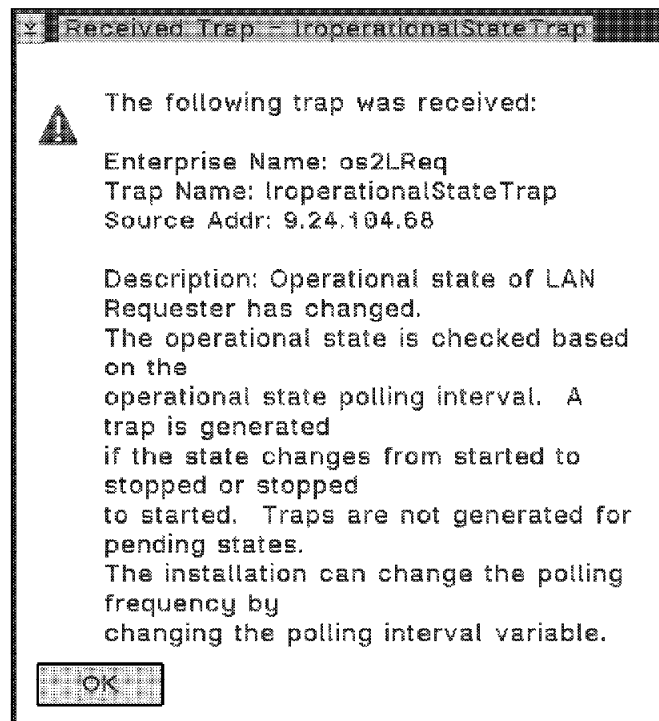


Figure 95. Pop-Up Message Showing Operation State Change of LAN Requester

3.7 Host Connectivity

The connectivity from NetView for OS/2's managing station to NetView on the host (MVS, VM, or VSE) is provided by Communications Manager/2. For a detailed description of how Communications Manager/2 is set up to enable this connection in an LU 6.2 (APPC) environment, please see 9.1, "Connecting to NetView for MVS/ESA" on page 189.

You can use the NetView for OS/2 Host Connection program to relay information to host NetView about events occurring on your LAN by converting selected events and forwarding them to the host NetView program where they can be displayed on your NetView console.

The *IBM NetView for OS/2 User's Guide, SC31-8099* gives a good overview of what the NetView for OS/2 Host Connection program can do. For an account of our experiences with setting up the Host Connection program including setting up Communications Manager/2, creating event filters, using host NetView to monitor SNMP events, and using the host NetView RUNCMD command to initiate appropriate responses down to the NetView for OS/2 managing station, please see 9.1, "Connecting to NetView for MVS/ESA" on page 189.

Chapter 4. OS/2 Agents

NetView for OS/2 provides two methods of managing OS/2 workstations using SNMP. The first is by using the standard MIB tables of the SNMP agent, and the second is through using the private MIB tables supplied as part of the System Information Agent (SIA) SNMP subagent.

The SNMP agent provides general information about the hardware and software installed on the workstation. It does this using information stored in the standard host resources MIB variables.

The SIA subagent provides private MIB variables that are an extension of the information available in the standard host resources MIB. These extensions allow the monitoring of performance information as well as controlling the workstation.

The following sections give some examples of how an OS/2 workstation can be controlled or monitored using SNMP. These examples are intended to show the sort of information that can be obtained from the MIB variables. In these examples, the variables were displayed or changed using the NetView for OS/2 MIB Browser application.

4.1 SNMP Agent

The NetView for OS/2 SNMP agent allows the viewing of information related to workstation resources. The host resources MIB in particular, provides information on workstation setup, devices, and software.

The following sections give examples of the sort of information that could be useful to monitor. In each case, the MIB tree path to access the relevant variable is shown.

4.1.1 Storage

The storage variables of the host resources MIB provide information about the physical and logical storage (both RAM and disks) on a particular workstation. This information can be useful in monitoring the usage of memory, disks, and buffers. The following examples illustrate the sort of information that can be viewed.

The following is the tree structure to get to the MIB variables shown:

```
* mgmt
  mib-2
    host
      hrDevice
```

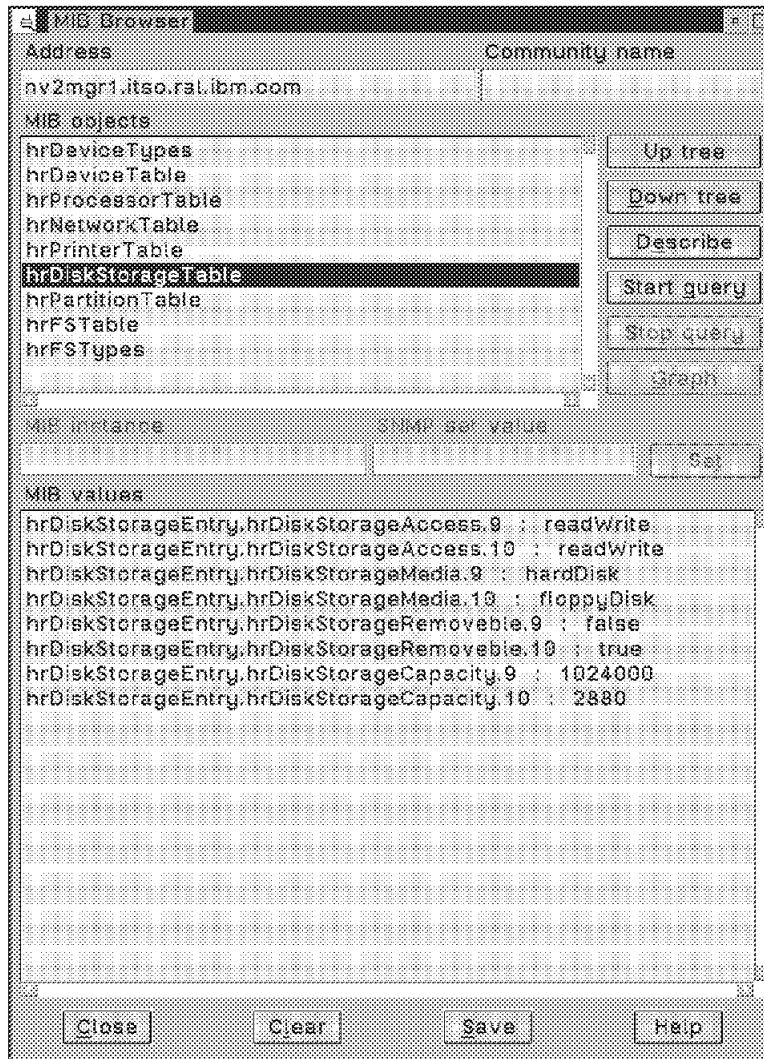


Figure 96. Physical Storage

The *hrDiskStorageTable* variable lists all the physical storage devices on a workstation. You can see from this example that this particular workstation has one 1GB disk drive and one 2.88MB diskette drive.

The next three examples show the logical storage configuration for the same workstation as used in the previous example. The MIB variables shown are reached using the following tree structure:

```
* mgmt
  mib-2
    host
      hrStorage
        hrStorageTable
```

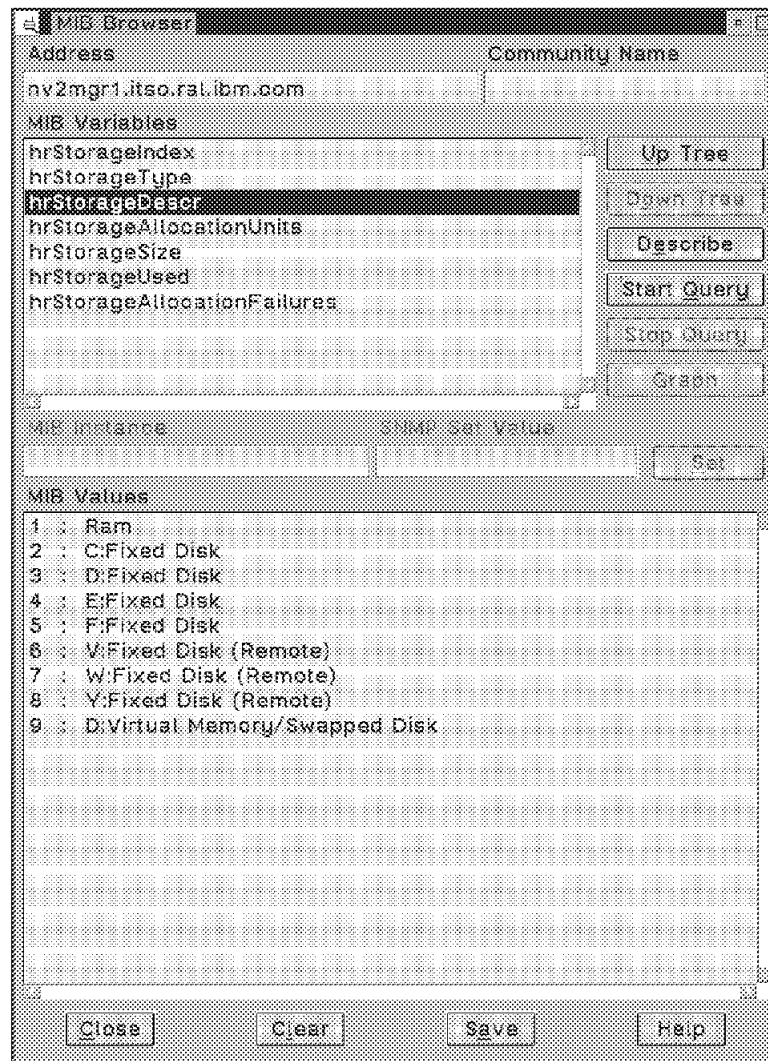


Figure 97. Logical Storage Description

The *hrStorageDescr* variable describes the logical storage entities on a workstation. The objects listed include the RAM size as well as the logical disk drives. The drives listed include LAN Server 3.0 remote drives; this MIB cannot see Novell NetWare logical drives.

Storage such as tapes and diskettes are also not seen by this variable as they are not typically allocated by the operating system to applications as logical volumes but rather as physical volumes.

In the example shown above, the workstation has four (local) logical disks and three network logical disks. It also shows that the D drive is being used by OS/2 for its swapper function.

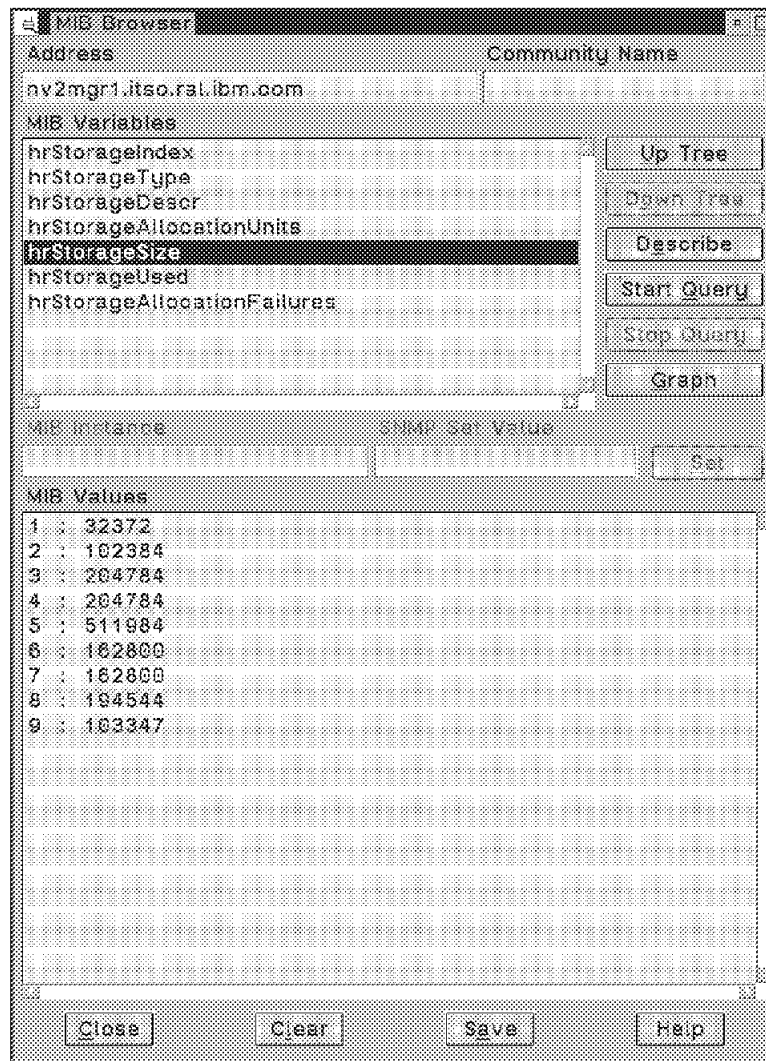


Figure 98. Logical Storage Size

The *hrStorageSize* variable displays the actual storage sizes for each of the entities listed in the *hrStorageDescr* variable for this particular workstation. This shows, for instance, that objects 3 and 4 are both 200MB logical disks. The value reported for the ninth object shows the amount of virtual memory available for use by applications, plus the current allocated SWAPPER.DAT file size.

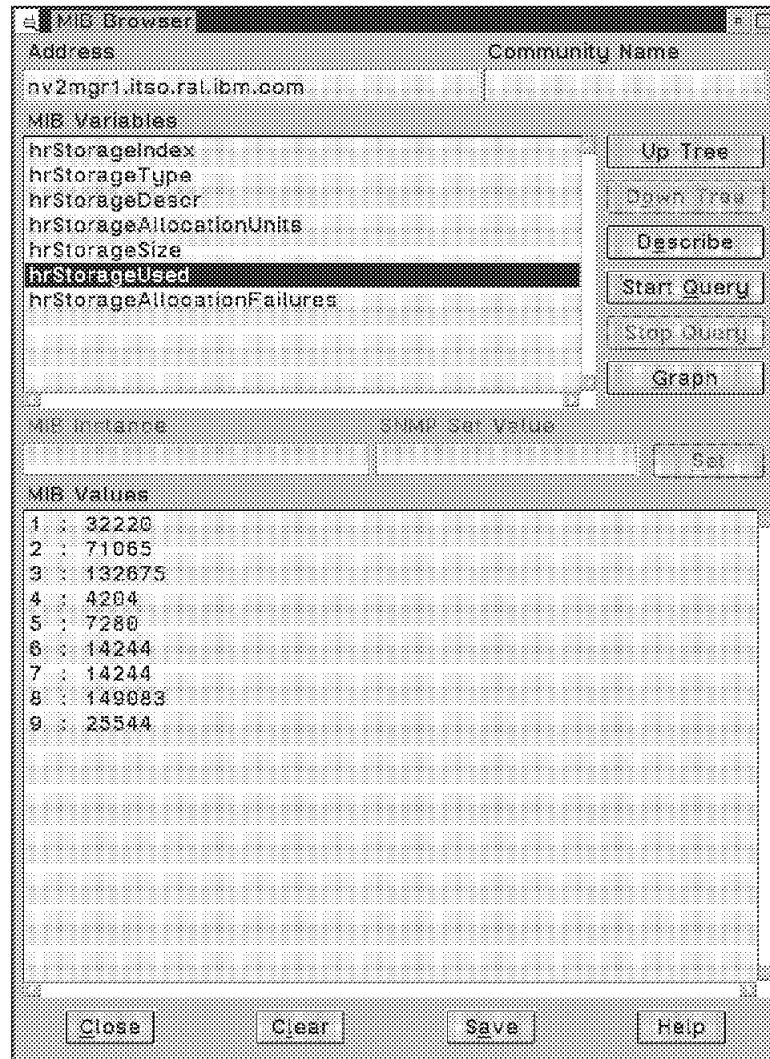


Figure 99. Logical Storage Usage

The *hrStorageUsed* variable displays the amount of storage in use by each of the entities listed in the *hrStorageDescr* variable. This usage relates to how the application, rather than the operating system, views storage. The value shown for the ninth object reflects the actual size of the SWAPPER.DAT file that is being used by OS/2 for its swapper function during this time slice.

4.1.2 Configuration

The host resources MIB also provides configuration information for the workstation. The following two examples show the sort of information that is viewable, together with how device status can be monitored.

The tree structure to get to the MIB variables shown is as follows:

```

* mgmt
  mib-2
    host
      hrDevice
        hrDeviceTable
          hrDeviceEntry

```

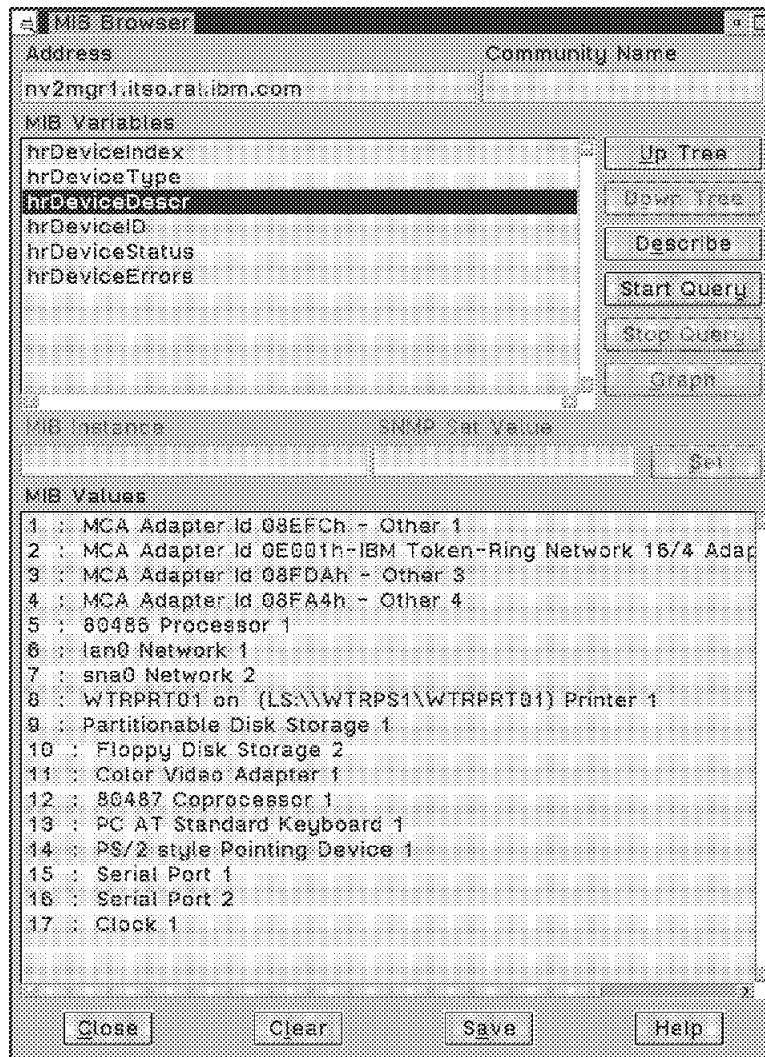


Figure 100. Device Description

This example shows information about the system configuration for this workstation as well as any adapters installed. The printer currently in use is a LAN-attached device.

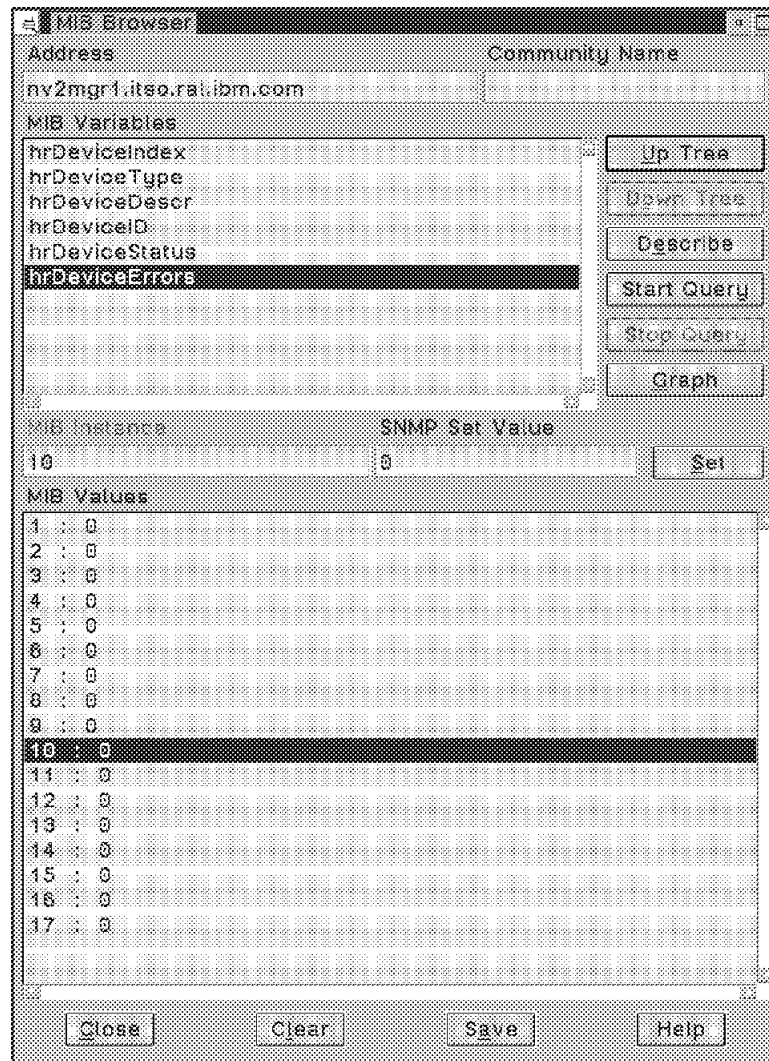


Figure 101. Device Errors

This example shows where errors will be logged for any of the devices listed in the *hrDeviceDescr* variable. The object numbers refer to the entities listed in the previous example.

This information could be a useful in providing an early warning that a particular device or feature requires preventive maintenance.

4.1.3 Printers

The following two examples show the sort of information available for workstation printers.

The tree structure to get to the variables shown is as follows:

```
* mgmt
  mib-2
    host
      hrDevice
        hrPrinterTable
          hrPrinterEntry
```

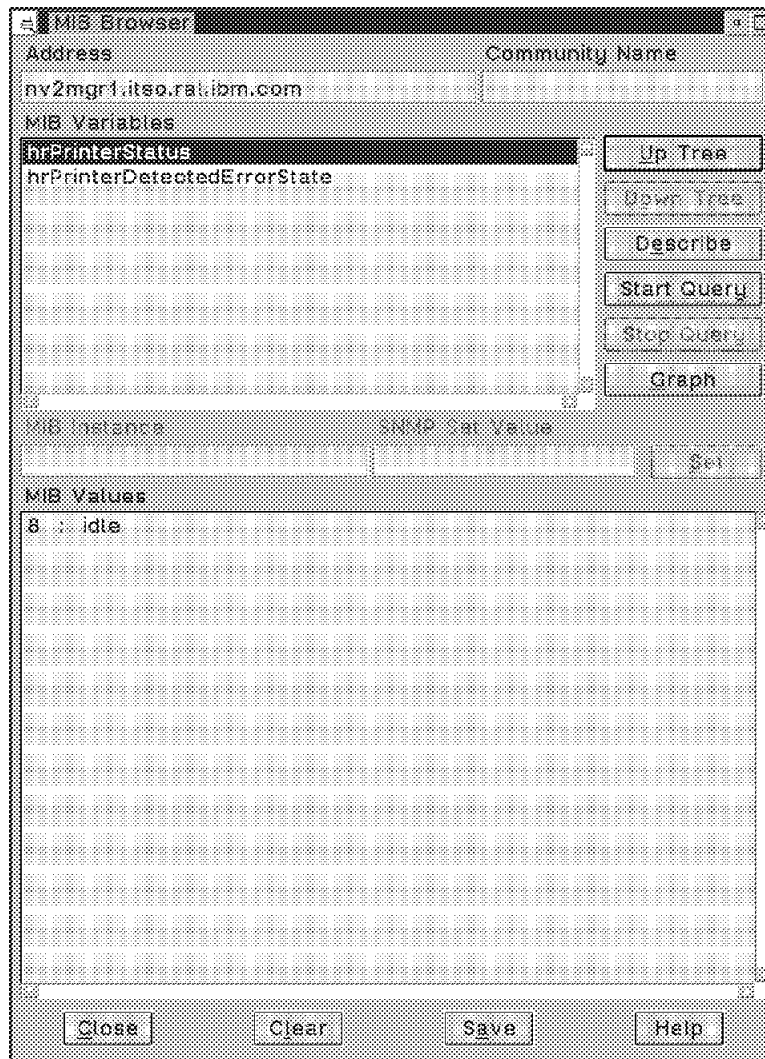


Figure 102. Printer Status

This example shows the current status of the printer allocated to workstation nv2mgr1. It could be local or remote (LAN attached). In this case the LAN-attached printer, as described by object 8 in the *hrDeviceDescription* variable, is idle. It is also possible for the status to be warmup, or unknown.

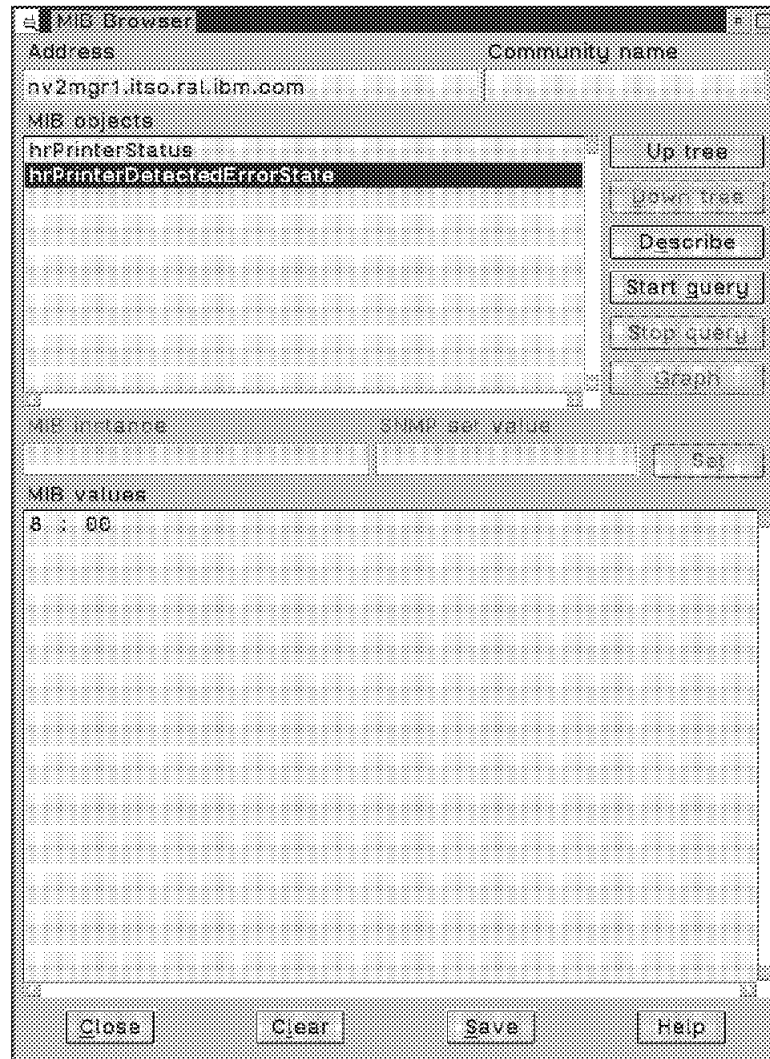


Figure 103. Printer Errors

The MIB in Figure 103 shows the variable which will get updated for any error conditions on the printer. In this case there are no errors but conditions such as low paper, paper jam, and low toner can be flagged. A complete list of conditions detectable can be found in *NetView for OS/2 Agents Guide - Appendix D*.

4.1.4 Software

The host resource MIB also gives information on the software configuration of a workstation. The next two examples show the software installed and actually running.

The tree structure to get to the variables shown is as follows:

```

* mgmt
  mib-2
    host
      hrSWInstalled
        hrSWInstalledTable
          hrSWInstalledEntry

```

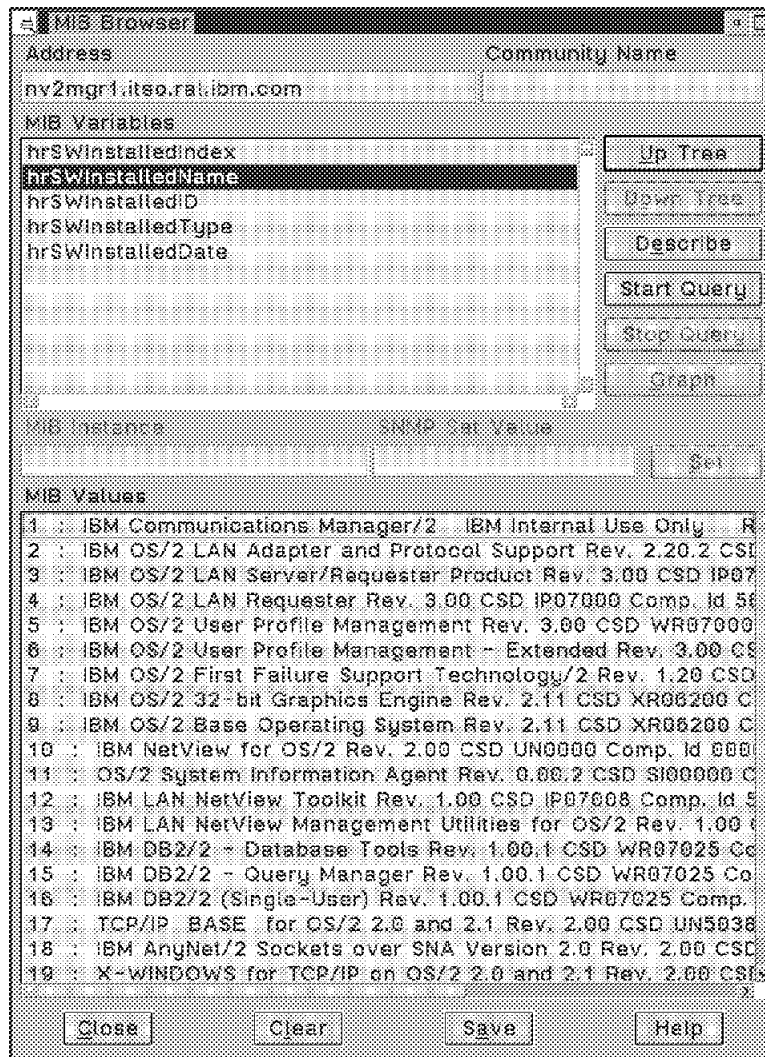


Figure 104. Software Installed

The *hrSWInstalledName* gives a complete listing of all IBM software running on a workstation. Information is taken from the relevant SYSLEVEL files and includes information on the software name, version, CSD level, and component identification.

The next example is reached by the following tree structure:

```

* mgmt
  mib-2
    host
      hrSWRun
        hrSWRunTable
          hrSWRunEntry

```

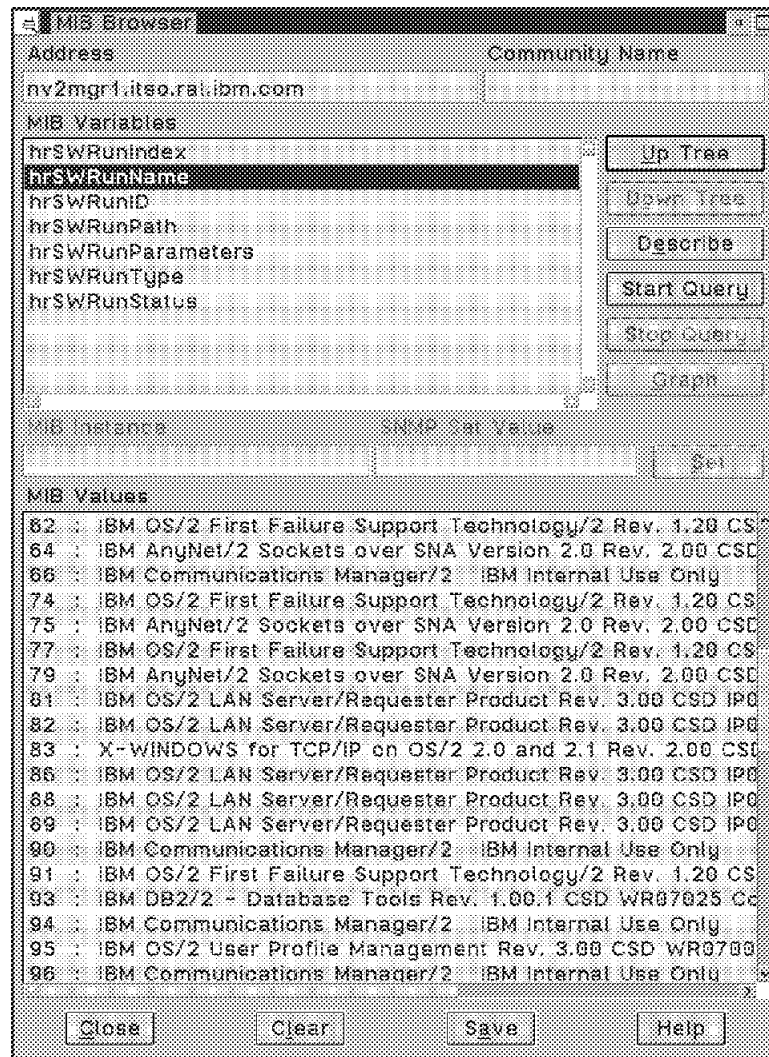


Figure 105. Software Running

The *hrSWRunName* variable lists all the IBM software actually running at the instant this variable was displayed. The information detail is the same as for the previous example of the *hrSWInstalledName* variable.

4.2 LMU MIB

NetView for OS/2 provides a private MIB that enables obtaining information from the LMU SNMP proxy agent. The information available from the proxy agent is:

- Workstations managed by LMU
- Vital product data
- Proxy agent setup
- Information about LMU itself

The first three examples are accessed using the following tree structure:

```
* private
  enterprises
    ibm
      ibmProd
        lmu
```

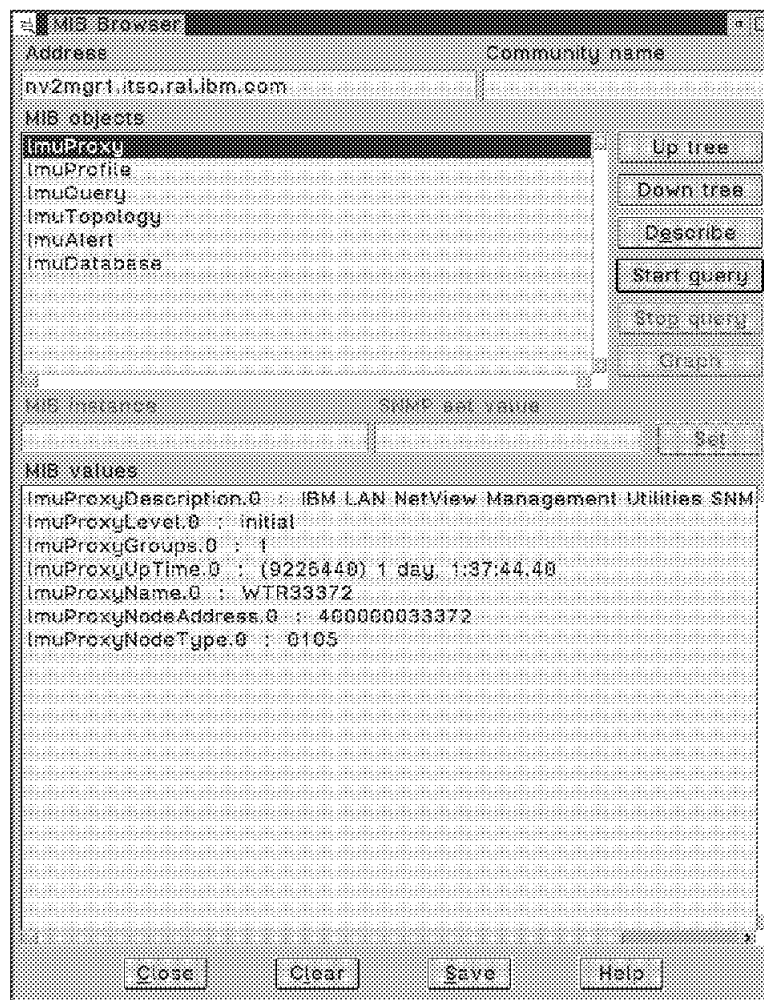


Figure 106. LMU Proxy Variable

The variables in the *ImuProxy* tree give information relating to the setup of the LMU SNMP proxy agent.

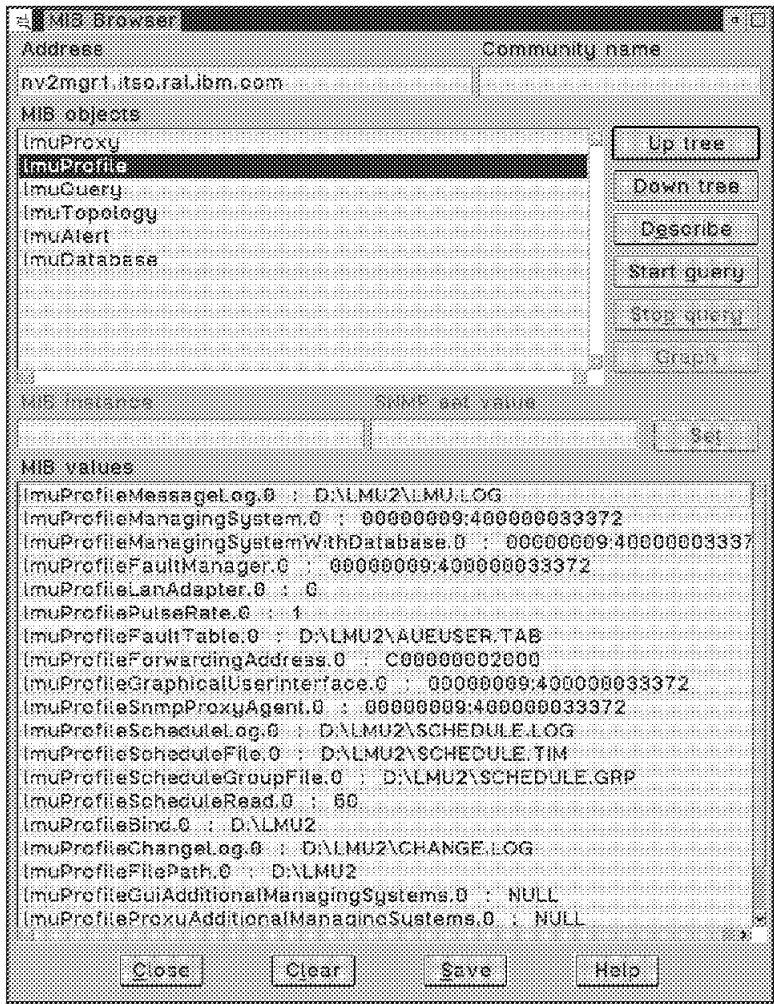


Figure 107. LMU Profile Variable

The variables under *ImuProfile* give details of the LMU profile setup (.CTL file) on the proxy agent workstation.

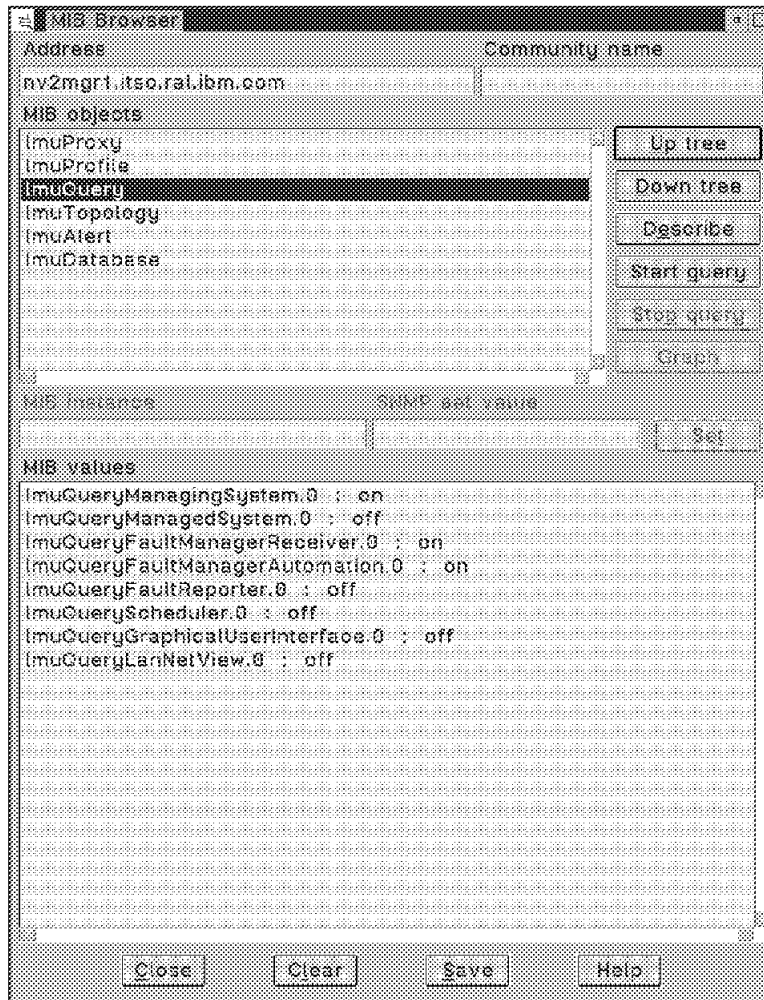


Figure 108. LMU Query Variable

The *ImuQuery* tree contains variables that show the status of the individual LMU components. In this case the LMU proxy agent workstation is also an LMU managing station.

The tree structure to get to the variables shown is as follows:

```
* private
  enterprises
    ibm
      ibmProd
        Imu
          ImuTopology
            ImuTopologyTable
```

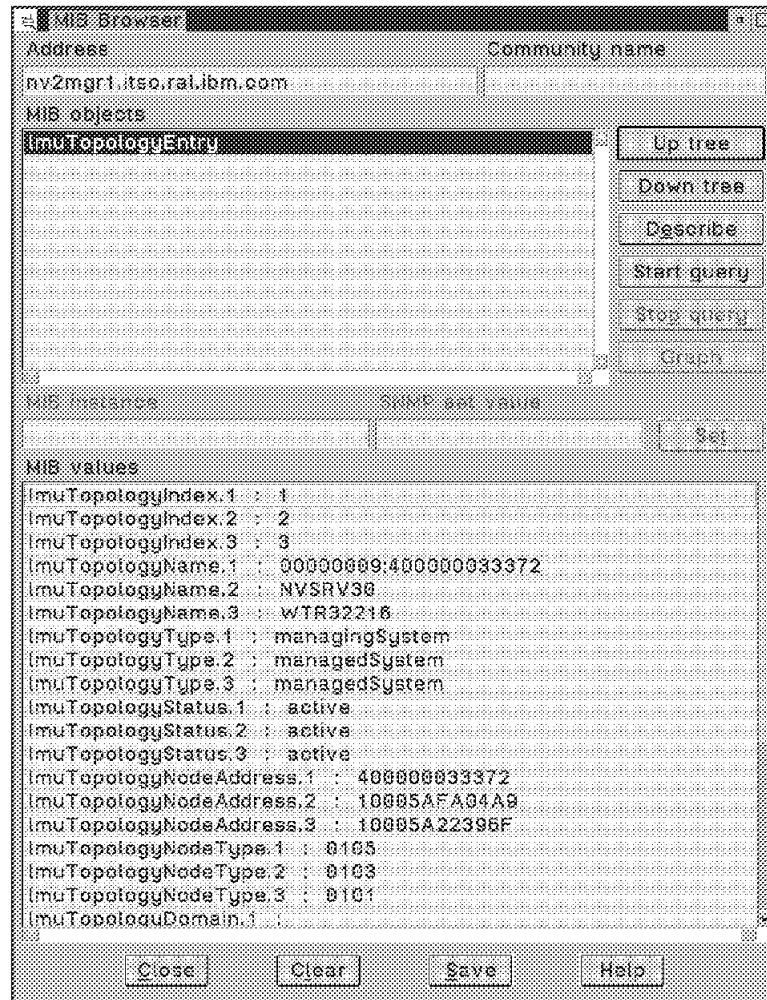



Figure 109. LMU Topology Entry Variable

The variables in the *ImuTopology* tree give details of the workstations currently being managed by LMU.

The tree structure to get to the variables shown is as follows:

```
* private
  enterprises
    ibm
      ibmProd
        Imu
          ImuAlert
            ImuAlertTable
```

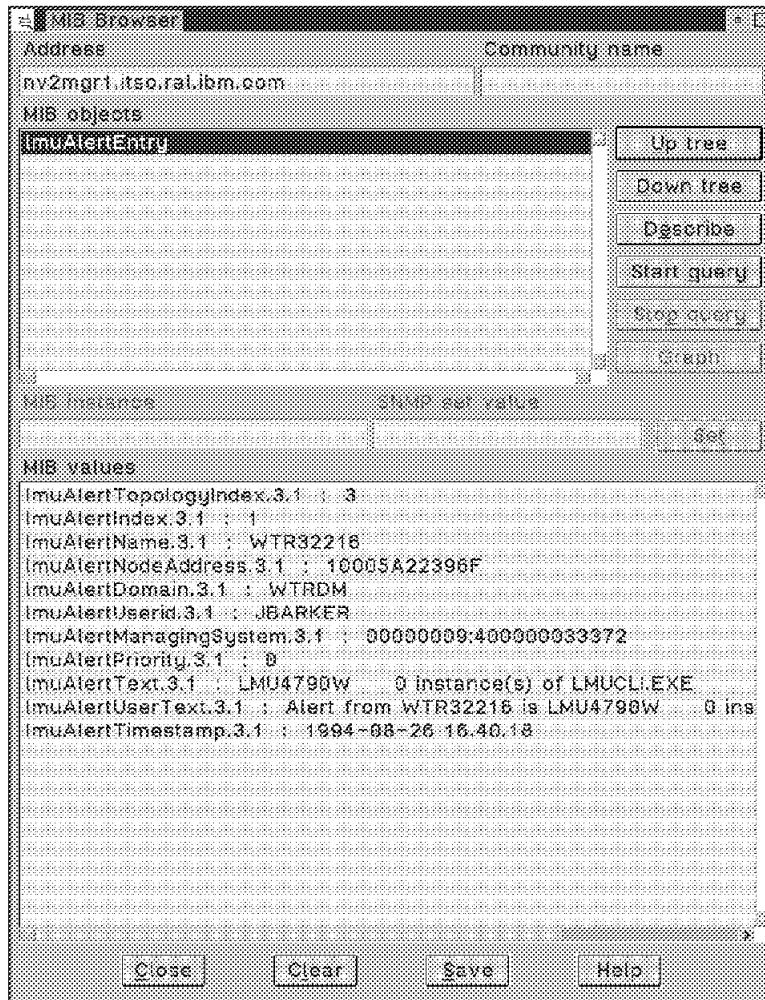


Figure 110. LMU Alert Entry Variable

The variables under the *ImuAlertEntry* tree give details of alerts issued by LMU-managed workstations. These alerts can optionally be sent to the NetView for OS/2 Event Displayer.

The remaining examples in the section all come from browsing variables that take information from the LMU database. This information is obtained by running the QUERYVPD.COM or QDOSVPD.COM command on the LMU-managed workstation.

The tree structure to get to the variables shown in the following three examples is as follows:

```
* private
  enterprises
    ibm
      ibmProd
        Imu
          ImuDatabase
            ImuDatabaseConfigurationTable
              ImuDatabaseConfigurationEntry
```

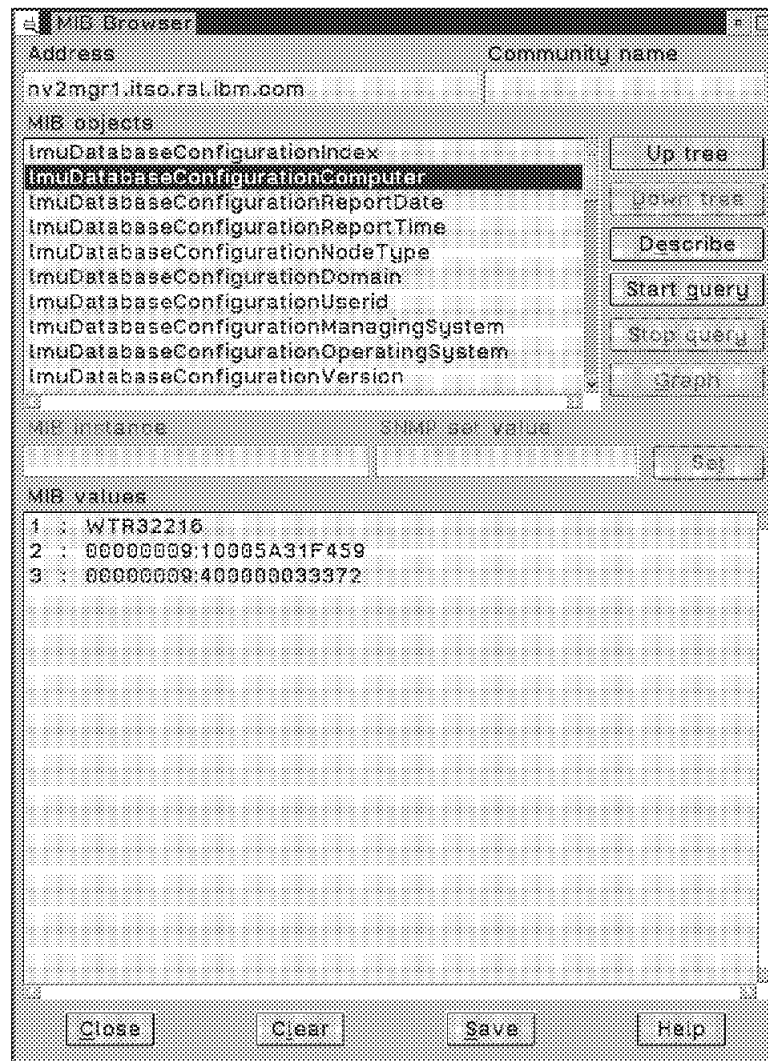


Figure 111. LMU Database Configuration Computer Variable

The *ImuDatabaseConfigurationComputer* variable lists all the LMU-managed workstations that the LMU database has VPD information on.

The next two examples show the model name and memory installed for the workstations listed in the previous example.

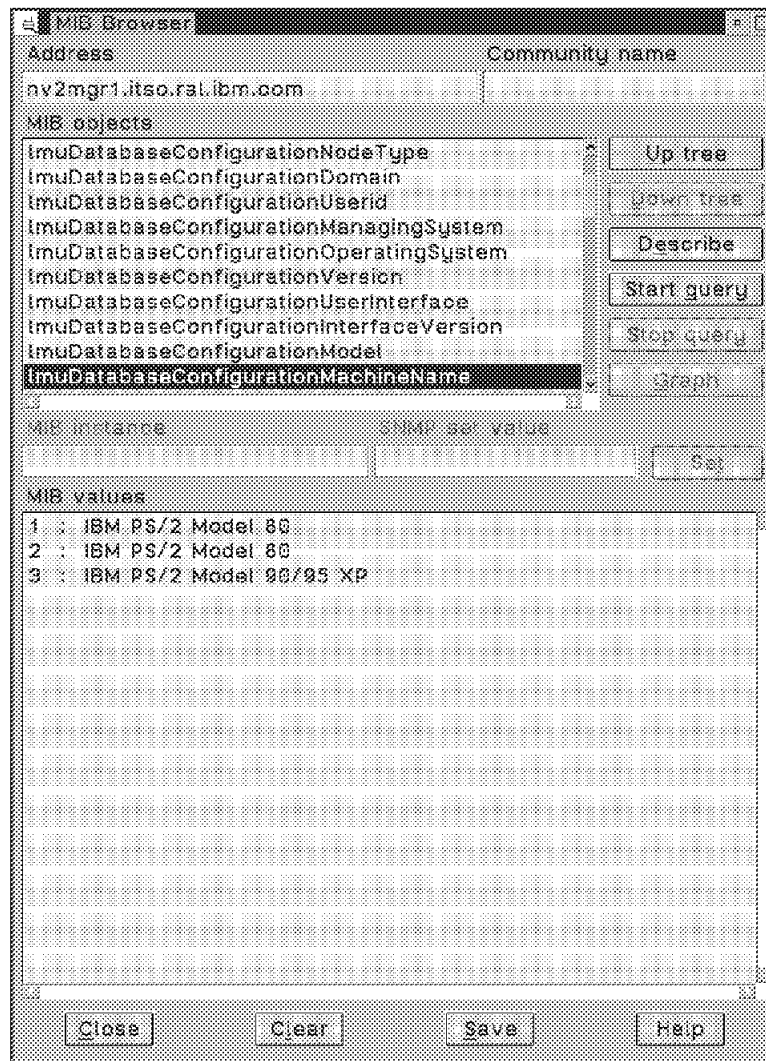


Figure 112. LMU Database Configuration Machine Name Variable

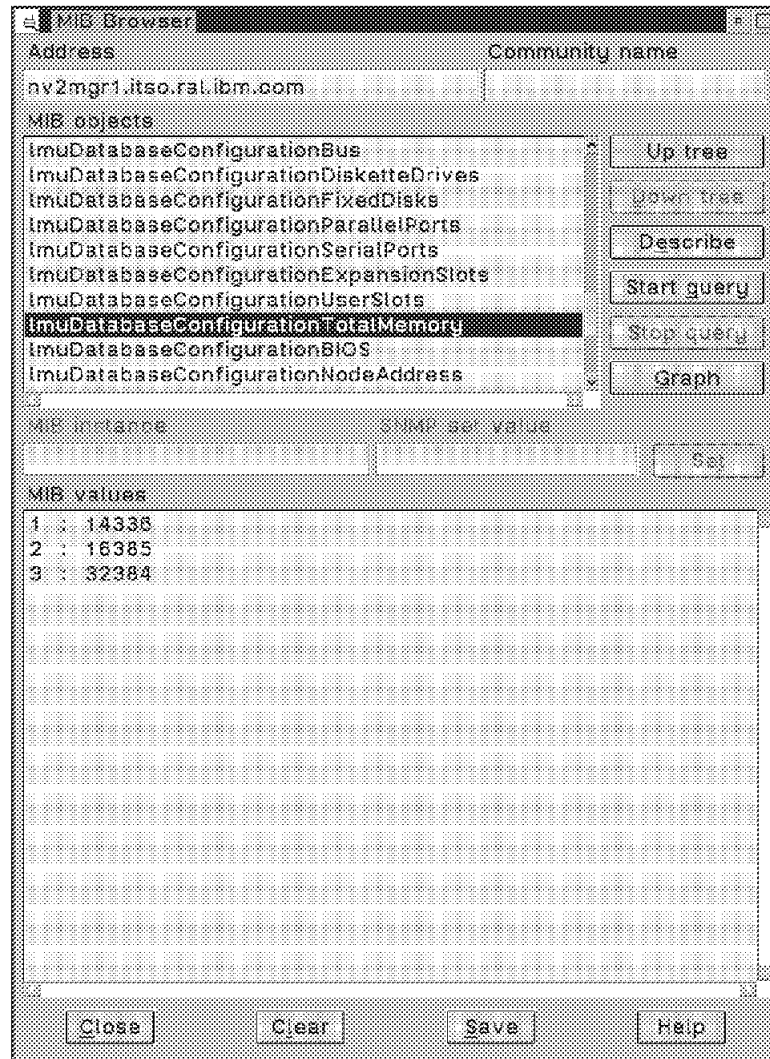


Figure 113. LMU Database Configuration Total Memory Variable

The next two examples show the sort of software information that is stored in the LMU database. The MIB tree to get to the variables shown is as follows:

```
* private
  enterprises
    ibm
      ibmProd
        Imu
          ImuDatabase
            ImuDatabaseSoftwareTable
              ImuDatabaseSoftwareEntry
```

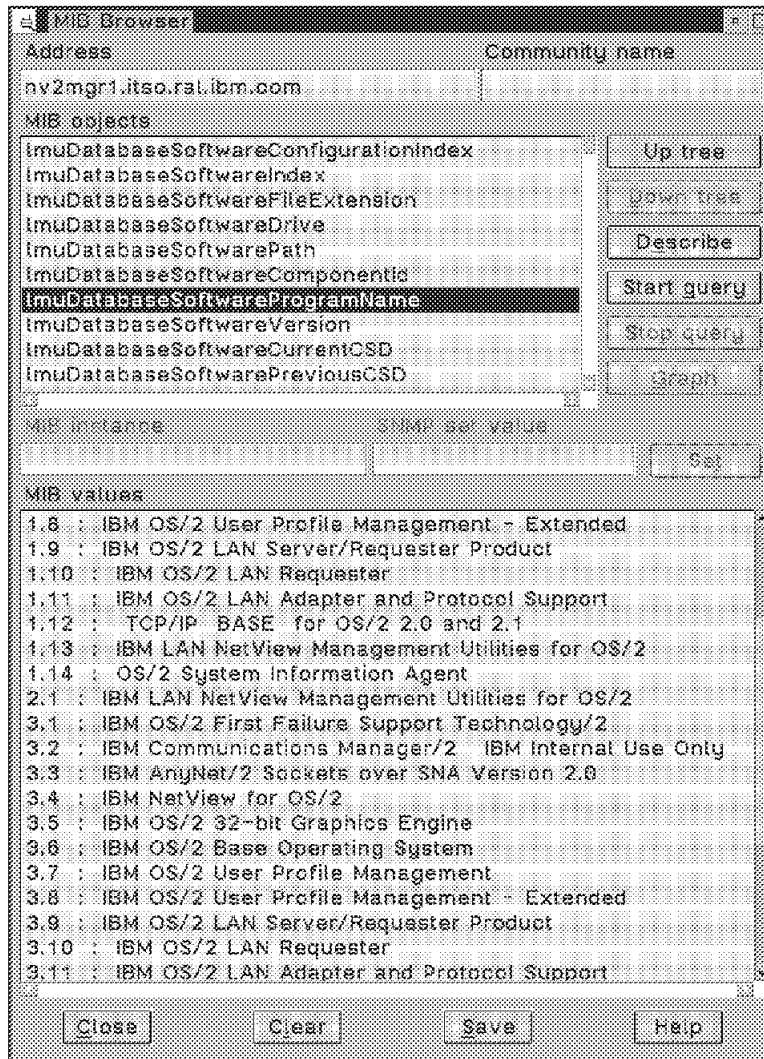


Figure 114. LMU Database Software Program Name Variable

This variable shows information on the software installed on the LMU-managed workstation. Unlike the information provided by the *hrSWInstalledName* variable of the SNMP host resources MIB, only the program name is supplied in this variable. More details are supplied using other variables, as shown in the next example where the CSD levels of the installed software are listed. It is possible to build a NetView for OS/2 MIB application that provides all the required information in one report.

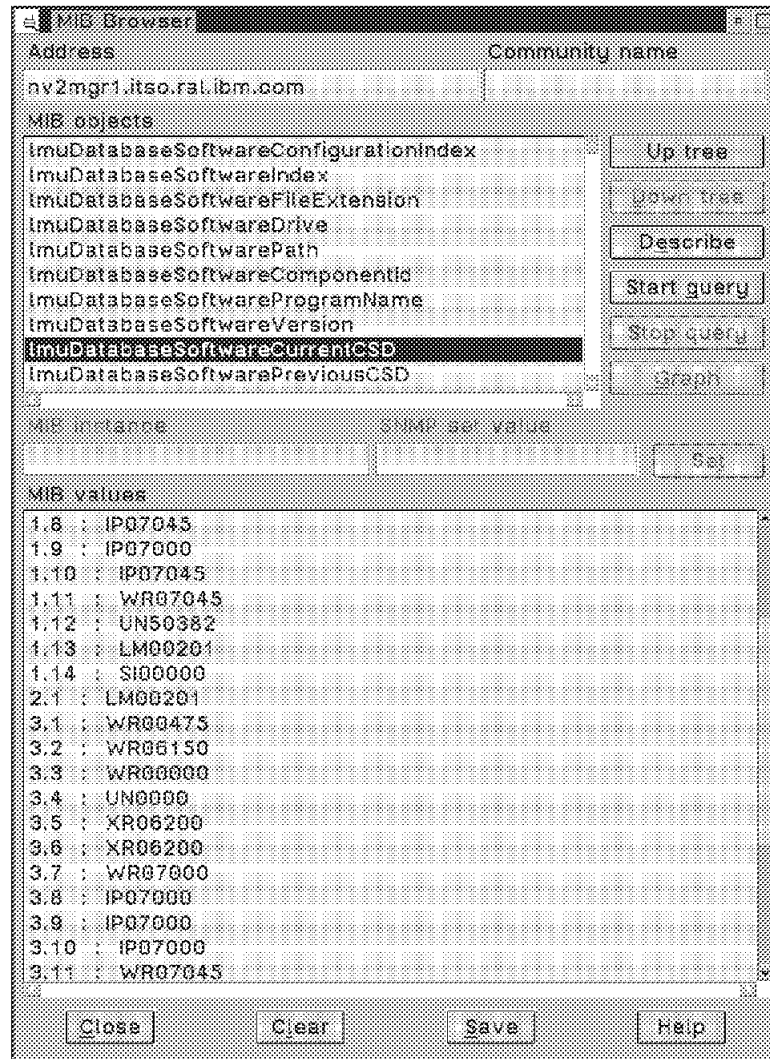


Figure 115. LMU Database Software Current CSD Variable

The final example in this section shows the sort of hardware VPD information that is available from the LMU database. The tree structure to get to the variables shown is as follows:

```
* private
  enterprises
    ibm
      ibmProd
        Imu
          ImuDatabase
            ImuDatabaseHardwareTable
              ImuDatabaseHardwareEntry
```

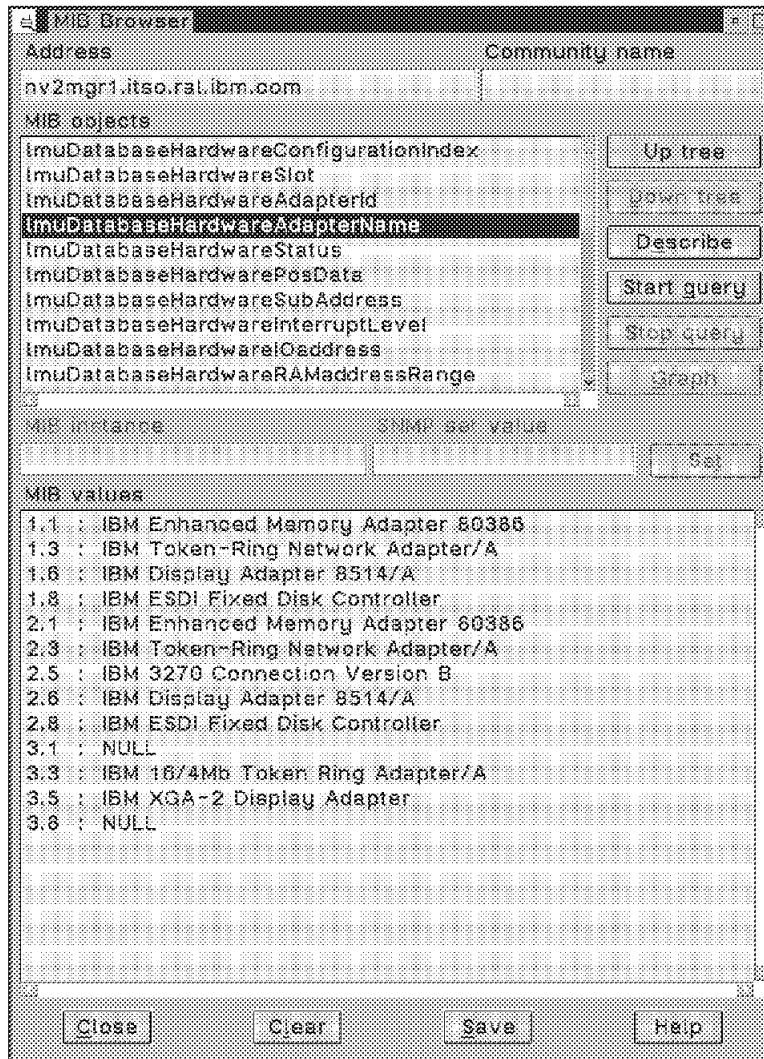


Figure 116. LMU Database Hardware Adapter Name

The NULL entries for object 3 instances 1 and 8 show the adapters in slots 1 and 8 on this particular machine were not identified when the VPD information was obtained. LMU provides a file (ADAPTERS.LST) where other vendor adapter information can be listed to ensure inclusion in the VPD report, and in the database.

The next example shows what the same information looks like when viewed using the DB2/2 Query Manager function.

Report			
COMPUTER	SLOT	ADAPTER ID	ADAPTER NAME
WTR32216	1	EDDF	IBM Enhanced Memory Adapter 80386
00000009:400030033372	1	8EDF	-
00000009:400030033372	3	E001	IBM 16/4Mb Token Ring Adapter/A
00000009:400030033372	5	8FDA	IBM XGA-2 Display Adapter
00000009:10005A31F459	1	EDDF	IBM Enhanced Memory Adapter 80386
NVSEV30	1	8EDF	IBM PS/2 SCSI Adapter with Cache
WTR32216	3	E000	IBM Token-Ring Network Adapter/A
WTR32216	6	EF7F	IBM Display Adapter 8514/A
WTR32216	8	EDDF	IBM ESDI Fixed Disk Controller
00000009:400030033372	3	8FA4	--
00000009:10005A31F459	3	E000	IBM Token-Ring Network Adapter/A
00000009:10005A31F459	5	E1EF	IBM 3270 Connection Version B
00000009:10005A31F459	5	EF7F	IBM Display Adapter 8514/A
00000009:10005A31F459	9	EDDF	IBM ESDI Fixed Disk Controller
NVSEV30	3	E001	IBM 16/4Mb Token Ring Adapter/A
NVSEV30	5	8FDE	IBM XGA Video Adapter

Figure 117. DB2/2 Query Manager

4.3 The System Information Agent (SIA) Subagent

NetView for OS/2 provides an SNMP subagent called the SIA. This subagent is an extension to the standard host resources MIB that allows control and monitoring of OS/2 V2.x running on a workstation. It provides information on performance and processes, a file monitor function, and the capability to shut down and reboot a workstation.

The following are examples of the sort of information available.

4.3.1 System Shutdown/Reboot

The tree structure to get to the variables shown is as follows:

```
* private
  enterprises
    ibm
      ibmProd
        os2SIA
          siaHostExtensions
            hrSystemExtensions
              hrSystemExt
                siaSystemGeneral
                  siaSysGenCommon
```

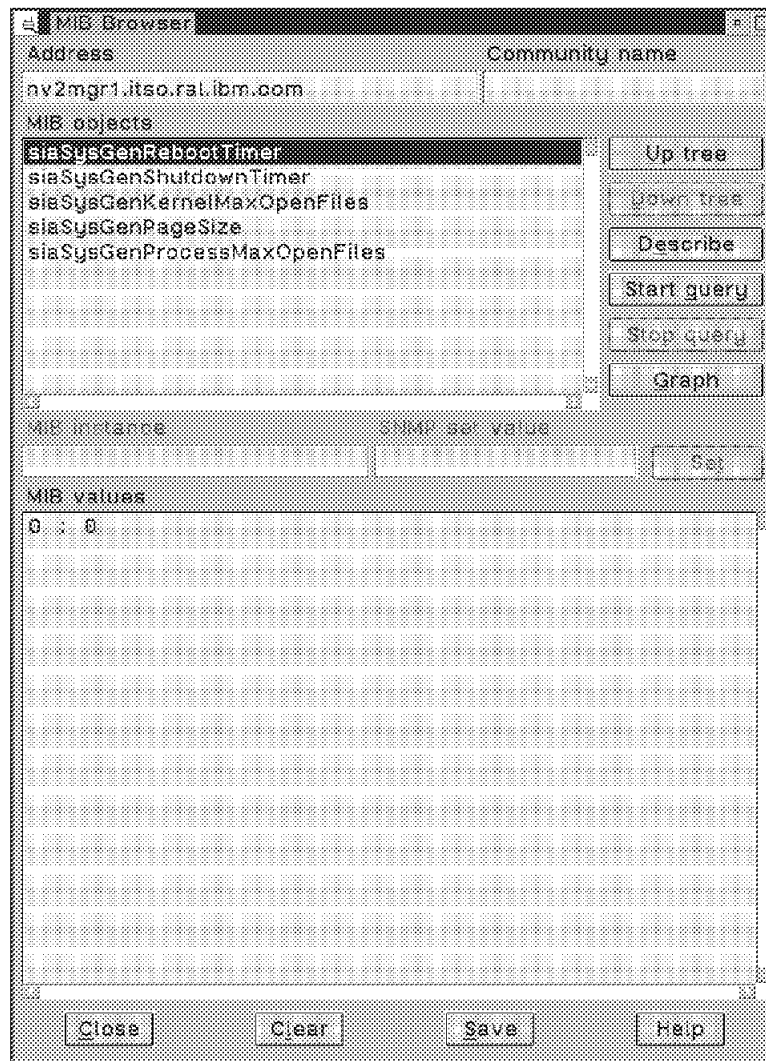


Figure 118. Reboot System

The *siaSysGenRebootTimer* variable is a read/write variable that sets the time before a workstation will be rebooted. Setting a value for the variable starts the reboot process.

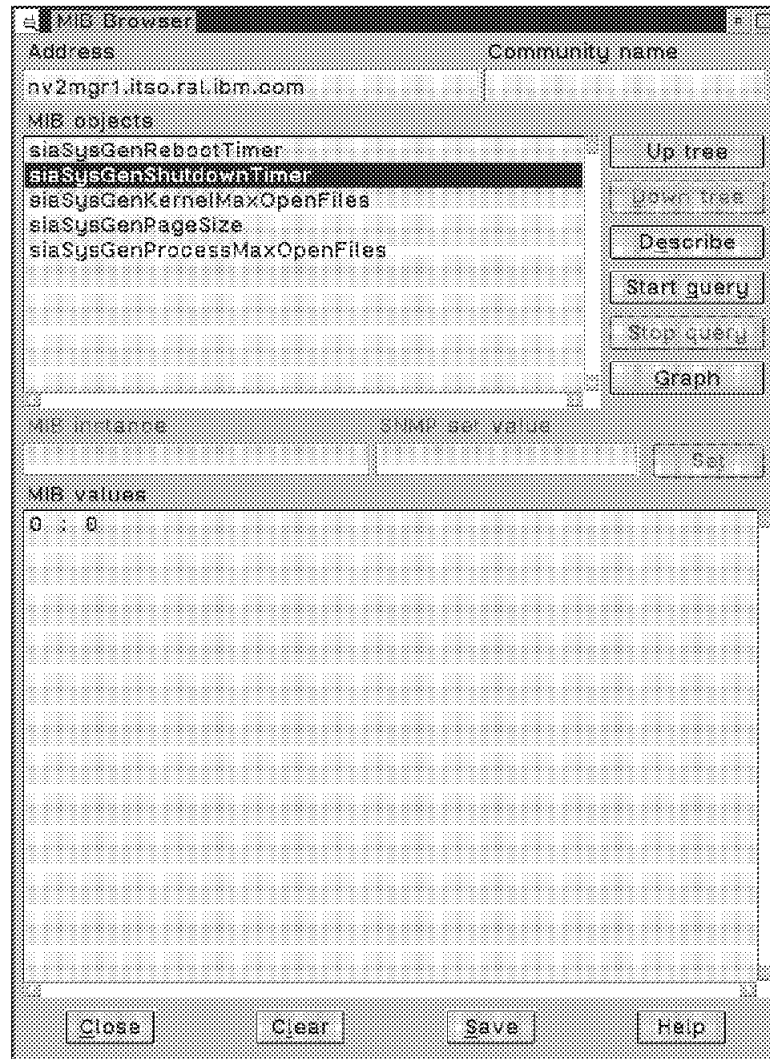


Figure 119. Shutdown System

The *siaSysGenShutdownTimer* is also a read/write variable, allowing the setting of an interval after which the target workstation will be shut down. Setting a value for this variable initiates a shut down.

4.3.2 Processor Utilization

The following three examples show how workstation processor utilization can be monitored.

The tree structure to get to the variables shown is as follows:

```

* private
  enterprises
    ibm
      ibmProd
        os2SIA
          siaHostExtensions
            hrDeviceExt
              siaProcessorUtil
                siaProcessorUtilCommon
                  siaProcessorUtilizationTable
                    siaProcessorUtilizationEntry

```

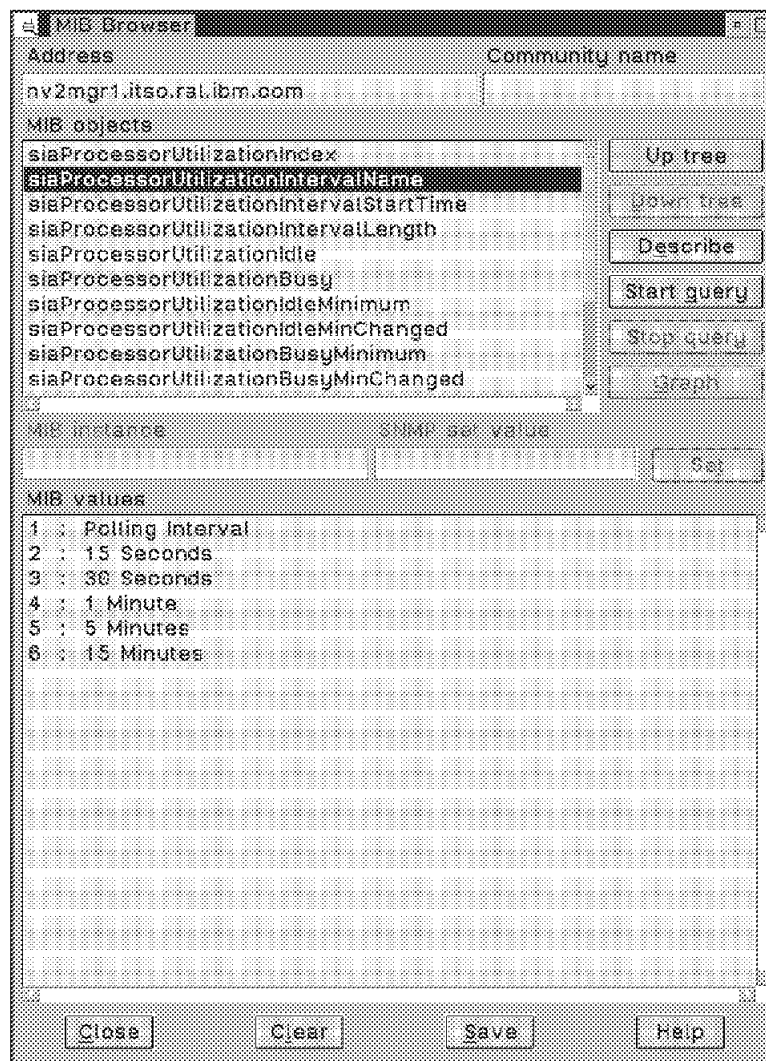


Figure 120. Interval Description

The *siaProcessorUtilizationIntervalName* variable information relates to the intervals in which the performance information was obtained. The first object is the length of time performance monitoring has been running.

The next two examples show processor busy idle and processor busy percentages for the period listed in the previous example.

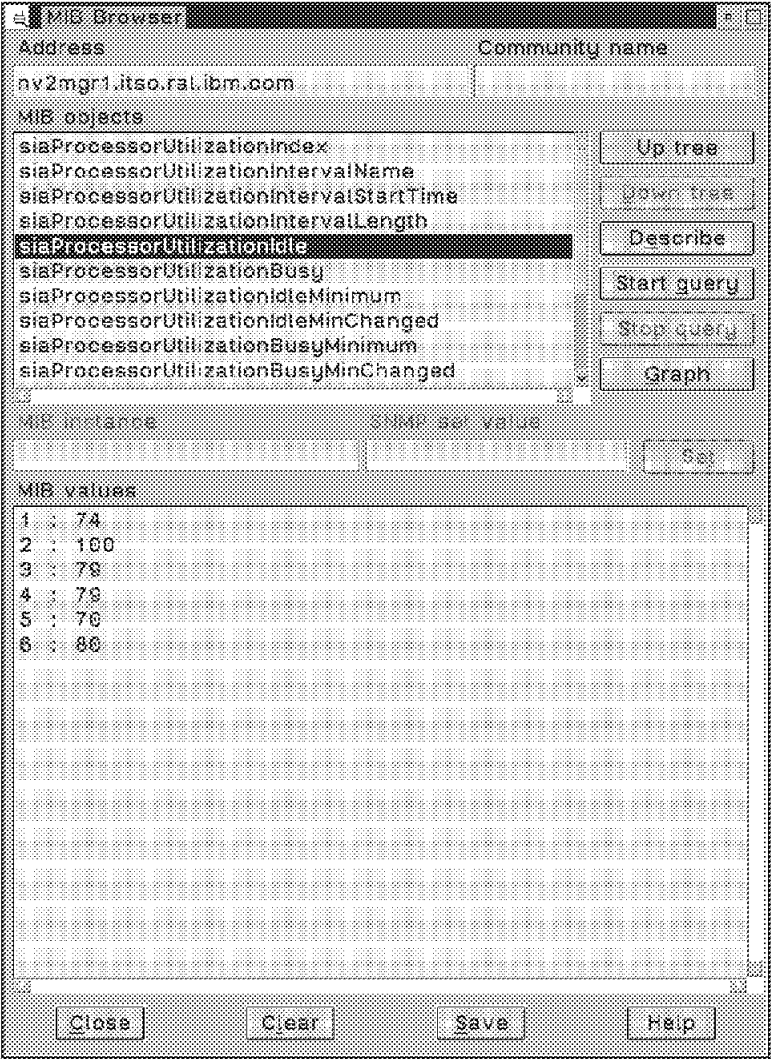


Figure 121. Processor Idle

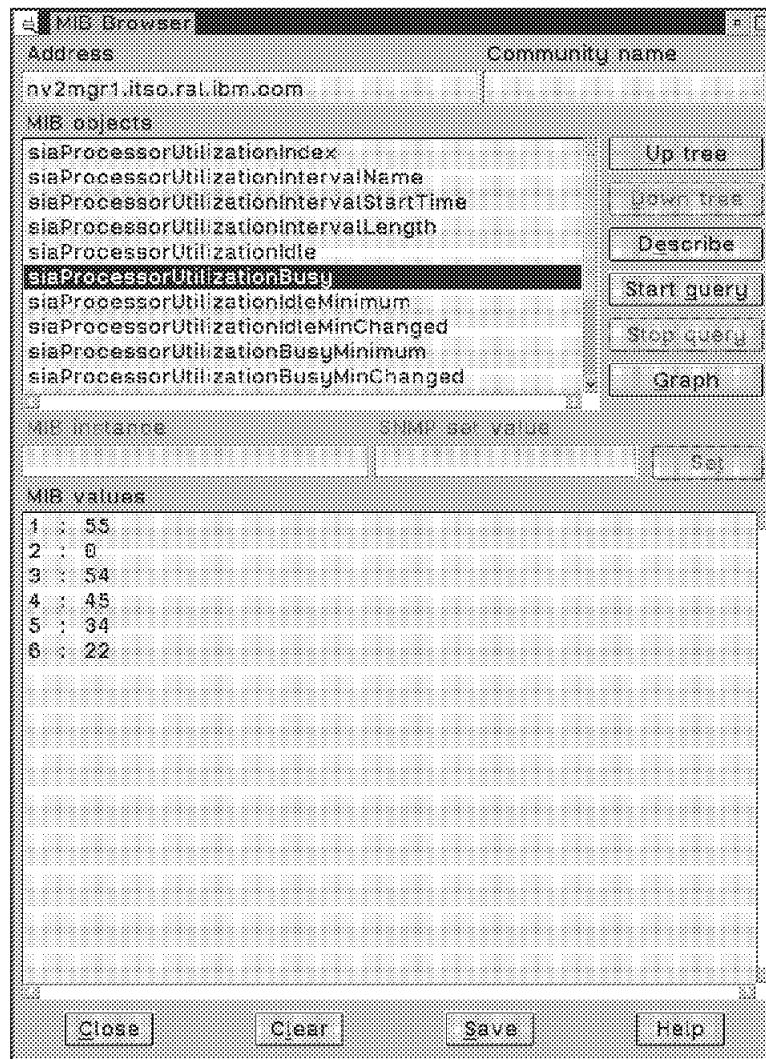


Figure 122. Processor Busy

4.3.3 Commands

The SIA subagent also provides MIB extensions that allow commands to be issued on OS/2 workstations. These commands are defined in the *sia_os2.cfg* file which can be found in the *c:\lanv2\etc\config* directory. The contents of the file look like this:

```
# SIA_BASE.CFG - Configuration file for OS/2 System Information Agent
#
siaProgramContact = "RTP IBM,Raleigh NC "
siaProgramControlNonExecCacheTime = 10
siaProgramControlPercentMultiplier = 100
siaProgramControlPollTime = 30
siaProgramControlFlags = 0
# Command table entries
siaCommandName·ECHO_GET" = "ECHO_GET"
siaCommandState = 3
siaCommandDescription = "Echo all the command GET environment variables for sysmond"
siaCommandGetStringAndParameters = "echo \"Hostaddress=%SIA_HOST_ADDRESS%, Hostname=%SIA_HOSTNAME% Hostbyname=%SIA_DOMAIN_NAME%
TimeOut=%SIA_COMMAND_TIME_OUT_VALUE% Owner=%SIA_COMMAND_OWNERID% Type=%SIA_COMMAND_RESULT_TYPE% InstanceId=%SIA_INSTANCE_ID%
InstanceName=%SIA_INSTANCE_NAME% Reason=%SIA_EXECUTION_REASON%\"""
siaCommandOutputResultIndex = 1
siaCommandName·ECHO_SET" = "ECHO_SET"
siaCommandState = 3
siaCommandDescription = "Echo the SET_VALUE command environment variable for sysinfoa"
siaCommandSetStringAndParameters = "echo \"SetValue=%SIA_COMMAND_SET_VALUE%\"""
siaCommandOutputResultIndex = 1
siaCommandName·PSTAT" = "PSTAT"
siaCommandState = 3
siaCommandDescription = "List all the processes running on this node"
siaCommandGetStringAndParameters = "pstat /c"
siaCommandOutputResultIndex = 1
siaCommandName·PING" = "PING"
siaCommandState = 3
siaCommandDescription = "Ping host from remote System Information Agent/2"
siaCommandSetStringAndParameters = "ping %SIA_COMMAND_SET_VALUE% 56 1"
siaCommandTimeOutValue = 5
siaCommandTimeToLive = 0
siaCommandOutputResultIndex = 1
siaCommandName·CONFIG" = "CONFIG"
siaCommandState = 3
siaCommandDescription = "View CONFIG.SYS file"
siaCommandGetStringAndParameters = "type c:\config.sys"
siaCommandTimeOutValue = 5
siaCommandOutputResultIndex = 1
siaCommandName·DEVICES" = "DEVICES"
siaCommandState = 3
siaCommandDescription = "Generate Installed Devices database"
siaCommandGetStringAndParameters = "blddevic.cmd"
siaCommandTimeOutValue = 5
siaCommandOutputResultIndex = 1
siaCommandOutputRowIndex = 0
siaCommandName·SOFTINS" = "SOFTINS"
siaCommandState = 1
siaCommandDescription = "Generate Installed Software database"
siaCommandGetStringAndParameters = "bldinstl.cmd"
siaCommandTimeOutValue = 15
siaCommandOutputResultIndex = 1
siaCommandOutputRowIndex = 0
```

The next two examples show the commands or routines currently defined in the *sia_cfg.cfg* file.

The tree structure to get to the variables shown in the next two examples is as follows:

```

* private
  enterprises
    ibm
      ibmProd
        os2SIA
          siaCommand
            siaCommandTable
              siacommandEntry

```

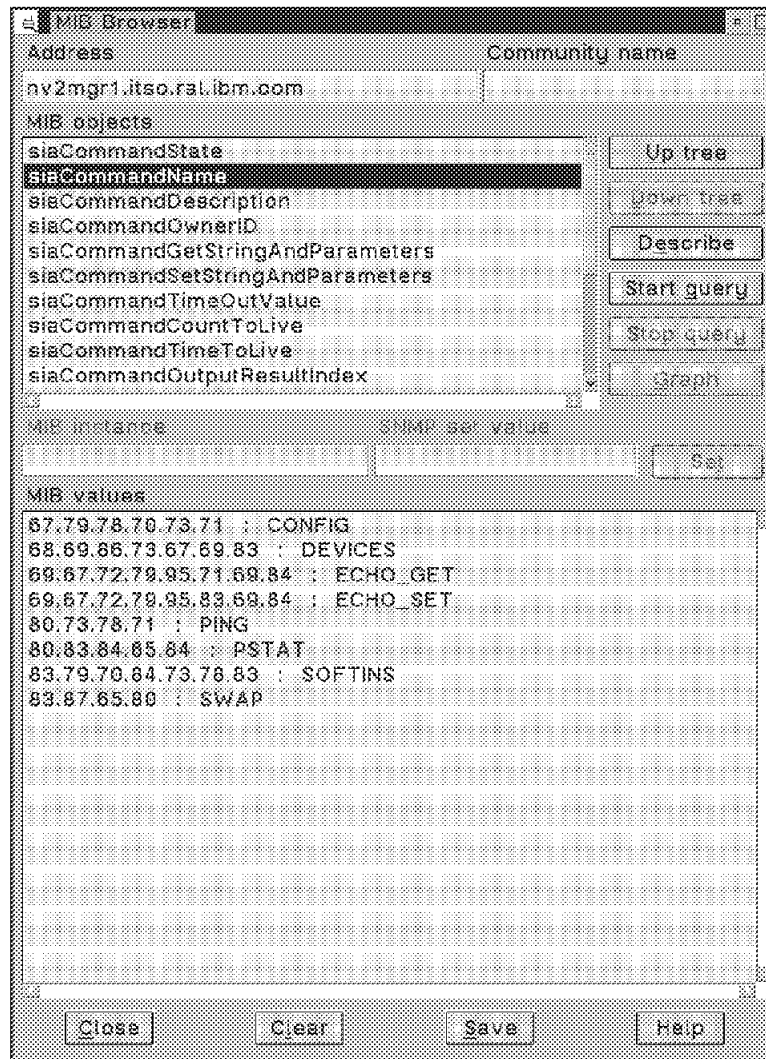


Figure 123. Command Names

The *siaCommandName* variable displays the routine names, as defined in the *sia_base.cfg* file.

The SOFTINS and DEVICES routines create a database of installed software and devices respectively. These databases can be viewed using the *mgmt*, *mib-2*, *host* MIB tree. These routines are started automatically as part of the OS/2 agent startup procedure, thus ensuring that the appropriate MIB variables contain information.

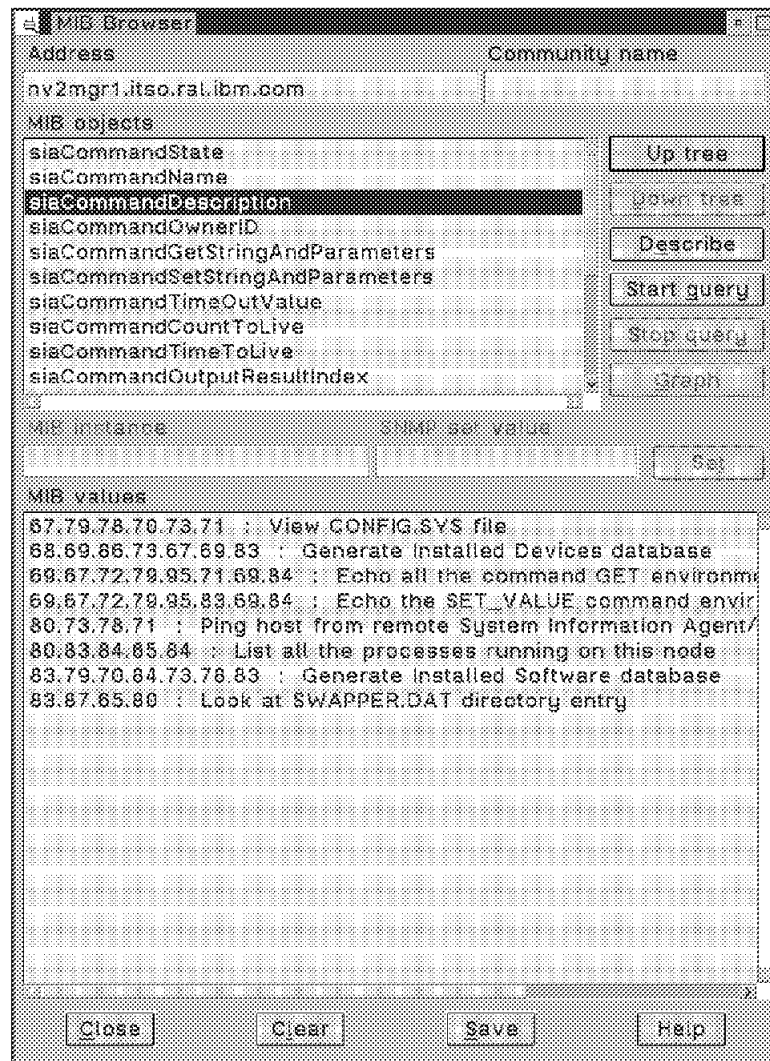


Figure 124. Command Description

The *siaCommandDescription* variable displays the descriptive text associated with the routines shown in the *siaCommandName* variable.

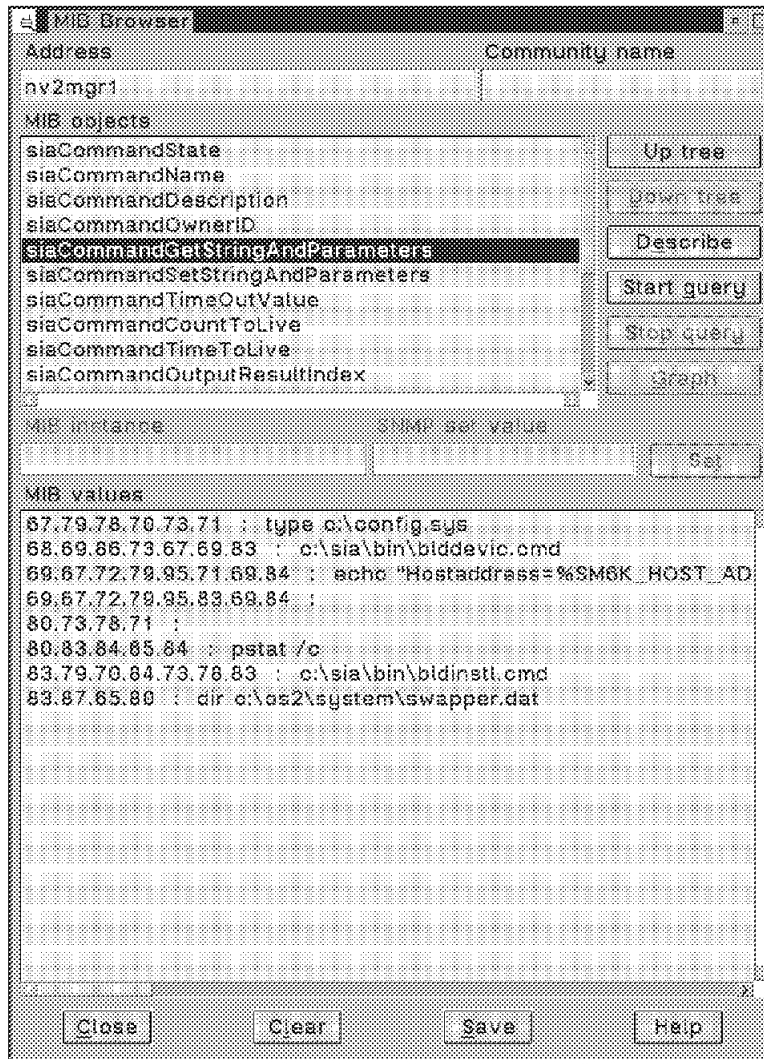


Figure 125. Command Definition

The *siaCommandGetStringAndParameters* variable displays the actual command names and options for the routines. These details can be updated by starting a query on the *siaCommandSetStringAndParameters* variable and then SETing the new values.

Important Note

You can only update existing commands using the MIB Browser application. New commands can be added by editing the *sia_base.cfg* file.

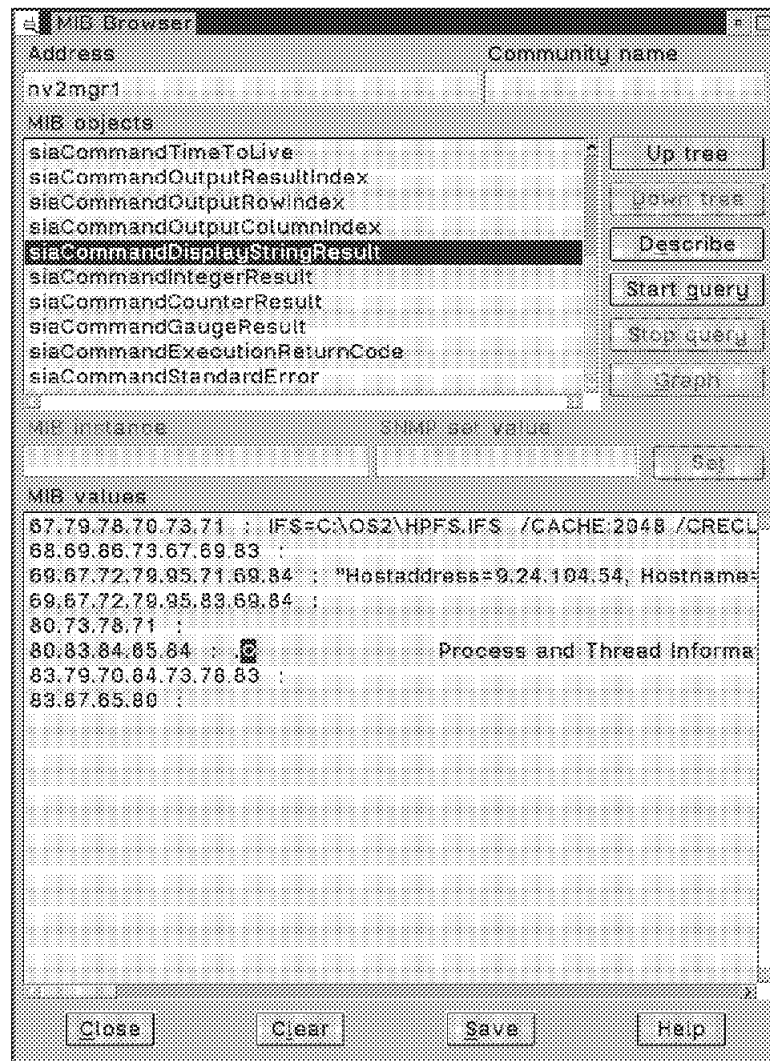


Figure 126. Command Results

In order to execute the defined routines, you start a query on the *siaCommandDisplayStringResult* variable. Any output from the routines is displayed in the MIB values window.

Chapter 5. LAN Server and LAN Requester Agents

NetView for OS/2 provides two SNMP subagents that allow management of the OS/2 LAN Server product.

The LAN Server and LAN Requester agent MIB groups contain such information as runtime configuration, statistics, performance, and vital product data. They also identify commands to control the Server and Requester and any associated traps generated from FFST alerts.

5.1 LAN Server Agent

The following sections show the type of information that can be obtained from the subagent MIB variables.

5.1.1 General Information

The *os2LANServer* group contains information on the LAN Server product and the state of the service.

The following is the tree structure to get to the MIB variable shown:

```
* private
  enterprises
    ibm
      ibmProd
        os2LS
          os2LANServer
```

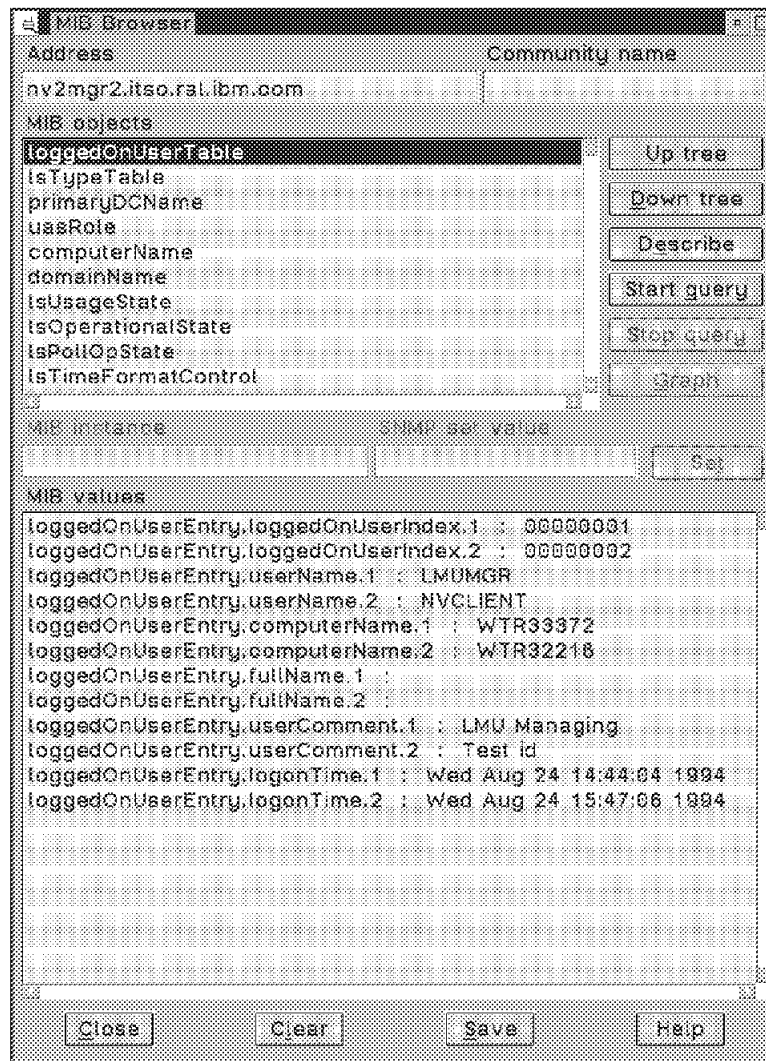


Figure 127. Logged On User Table

The *loggedOnUserTable* displays information about logged on users. This variable only contains information if this server is a primary or backup domain controller.

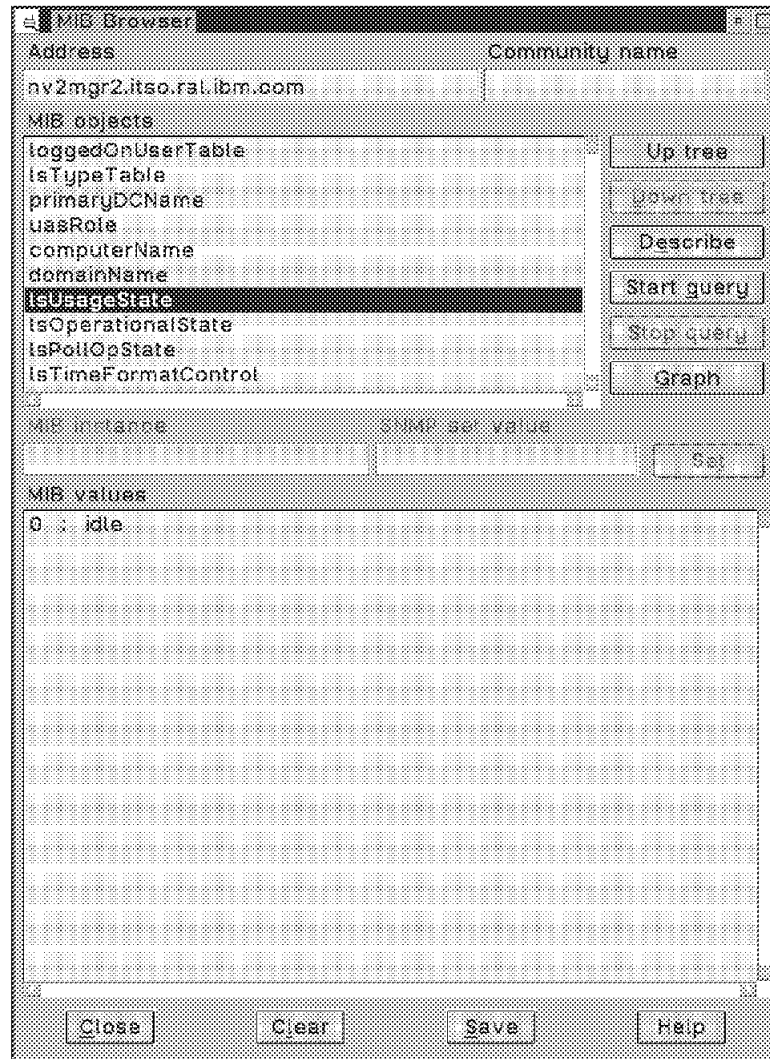


Figure 128. LAN Server Usage State

The *IsUsageState* displays whether LAN Server can accept client requests. A status of *idle* shows that it can accept requests. If the variable shows a status of *active* then at least one client is in session, while *busy* status shows that the server has reached its maximum number of connections or sessions.

5.1.2 Configuration

The *IsRuntimeConfig* group contains information on the runtime setup of LAN Server, as defined in the IBMLAN.INI file.

The following three examples show the sort of information available. All the variables shown are read/write, so values can be changed using the SET command.

The tree structure to get to the MIB variables is:

```

* private
  enterprises
    ibm
      ibmProd
        os2LS
          IsRuntimeConfig

```

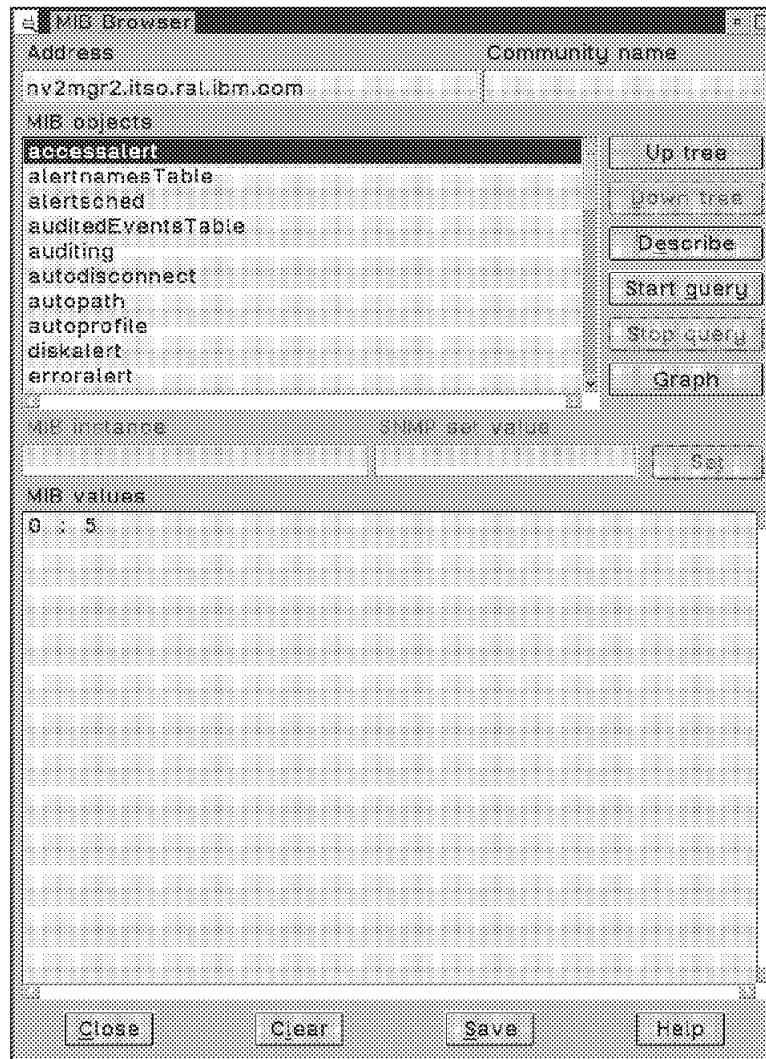


Figure 129. Access Alert

accessAlert is the number of invalid file accesses before an alert will be issued.

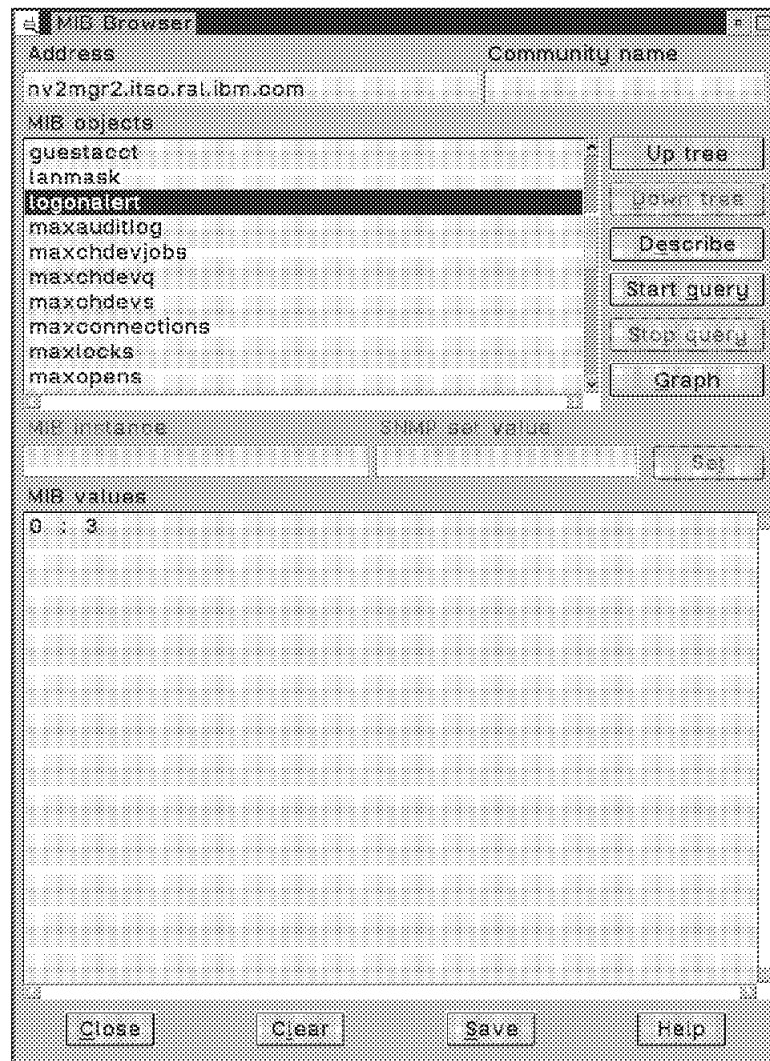


Figure 130. Logon Alert

logonalert specifies the number of invalid logon attempts before an alert will be issued.

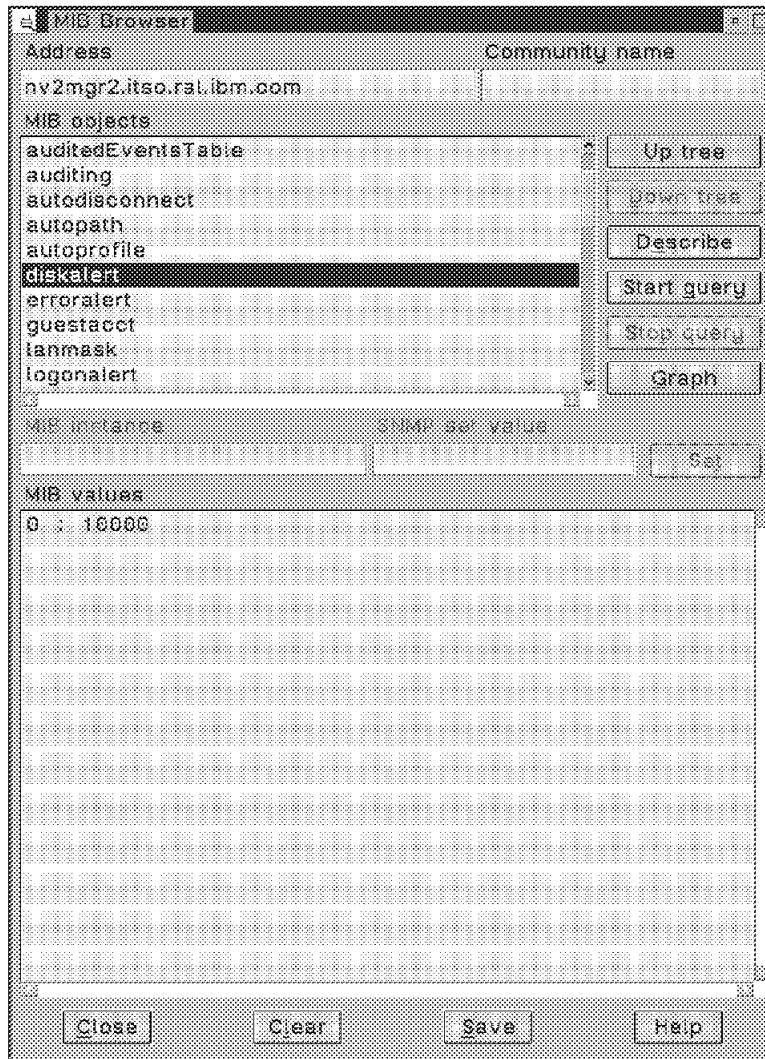


Figure 131. Disk Alert

diskalert specifies the threshold level (in kilobytes) before a disk free space alert will be issued.

5.1.3 Statistics

The *IsStatistics* group contains information on items that affect the performance of LAN Server. The next two examples show two MIB variables that could be useful to monitor.

The tree structure to get to the MIB variables is:

```

* private
  enterprises
    ibm
      ibmProd
        os2LS
          IsStatistics
  
```

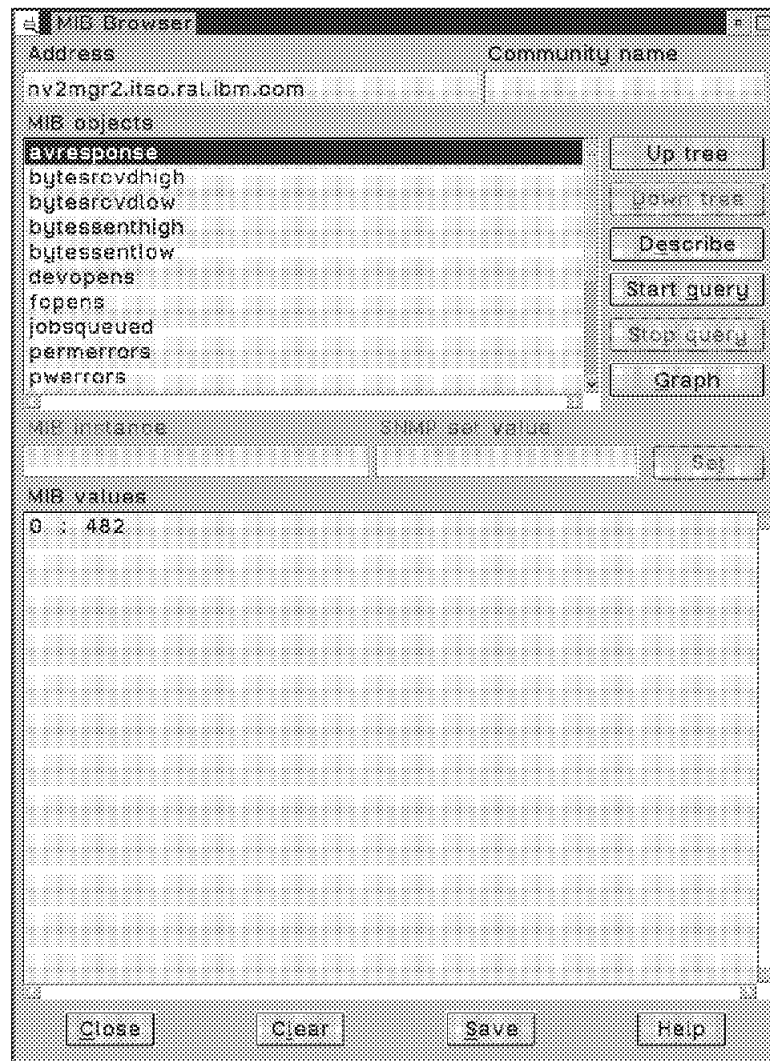


Figure 132. Average Server Response Time

avresponse shows the average server (internal) response time, in milliseconds.

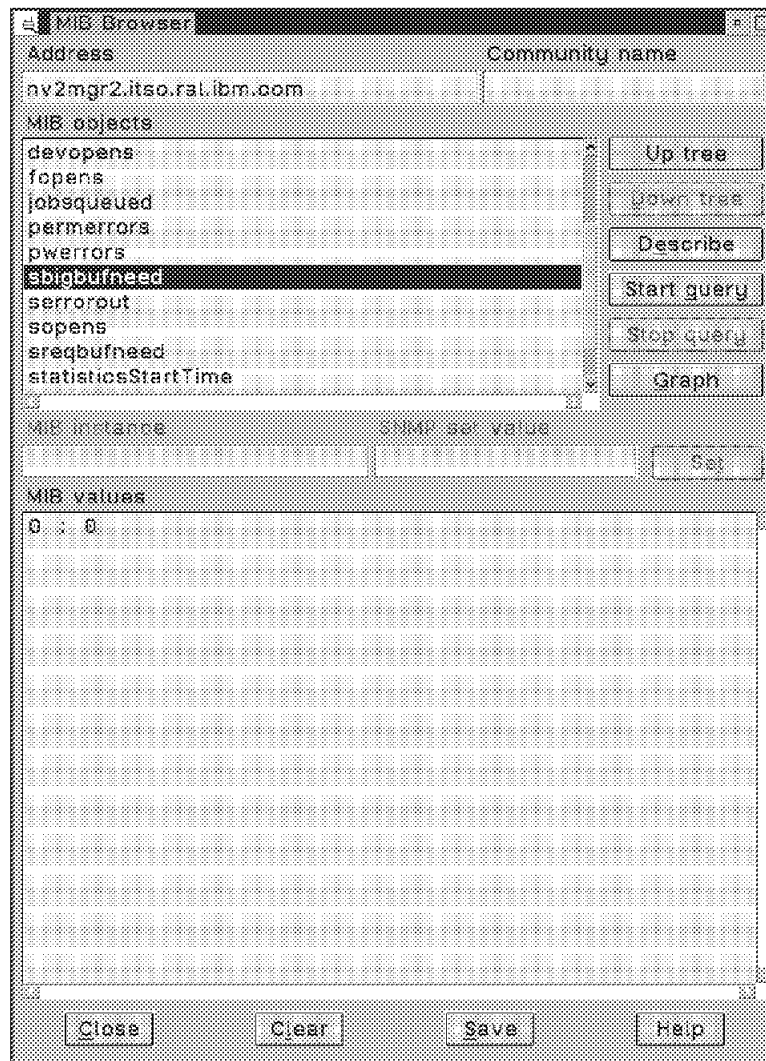


Figure 133. Failed Attempts to Allocate Big Buffer

sbigbufneed shows the number of times the server tried to allocate a big buffer and failed. LAN Server uses big buffers for large sequential file access operations, so a failure to obtain the buffers could adversely affect performance.

5.1.4 Performance

The *IsPerfMetrics* group contains 50 performance counters that start when the server starts. The polling interval for collecting this information can be changed using the *IsCollectStartTime* variable.

```
* private
  enterprises
    ibm
      ibmProd
        os2LS
          IsPerfMetrics
```

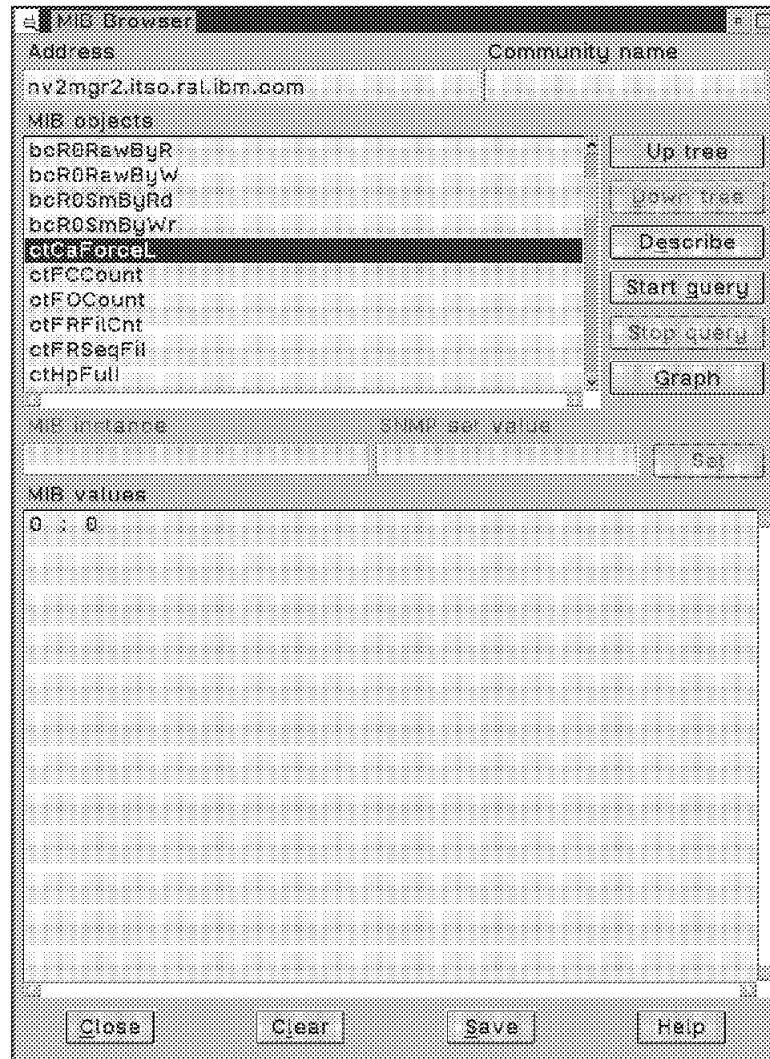


Figure 134. Server Cache Flushed

The *ctCaForceL* variable lists the number of times the server cache was flushed, due to no free buffers being available. A high number of cache flushes can greatly increase program loading and file transfer times.

5.2 LAN Requester Agent

The following examples show the sort of information that can be obtained using the subagent MIB tree. We have tried to show areas of interest rather than going into great depths for a particular subject or agent.

5.2.1 General Information

The *ibmLANIniConfig* group contains objects that monitor and control the IBMLAN.INI configuration file. You can back up and restore the file, as well as display the time and date of the current and backup files.

The first example shows how the *ibmLANIniFileName* variable displays the current active configuration file name. This could be useful in checking if users are using their own customized file name.

The tree structure to get to the variable shown is as follows:

```
* private
  enterprises
    ibm
      ibmProd
        os2LReq
          ibmLANIniConfig
```

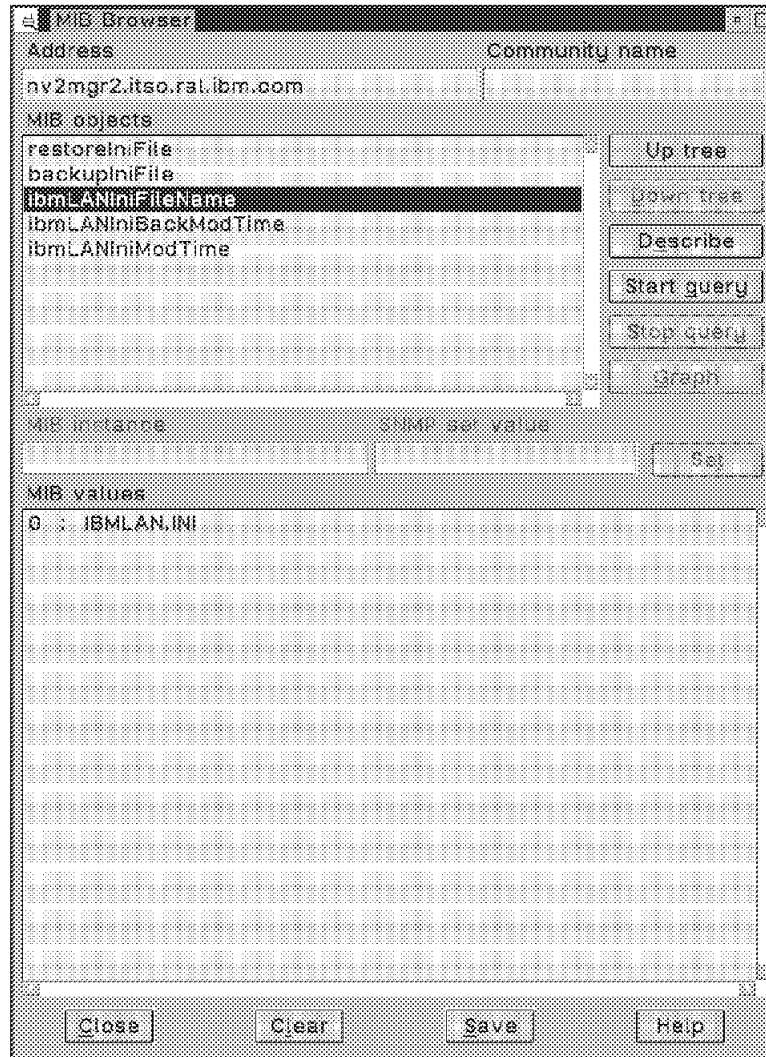


Figure 135. IBM LAN.INI File Name

5.2.2 Configuration

The *IrRuntimeConfig* group defines the runtime characteristics of LAN Requester as defined in the IBMLAN.INI file. By doing a GET or SET on a particular variable, the setup of LAN Requester can be altered. The next example shows the contents of the *wrkHeuristics* variable. The 34-digit display refers to fine tuning options that can be set. The exact meaning of each digit is explained in *LAN*

The following is the tree structure to get to the MIB variable shown:

```
* private
  enterprises
    ibm
      ibmProd
        os2LReq
          IrRuntimeConfig
```

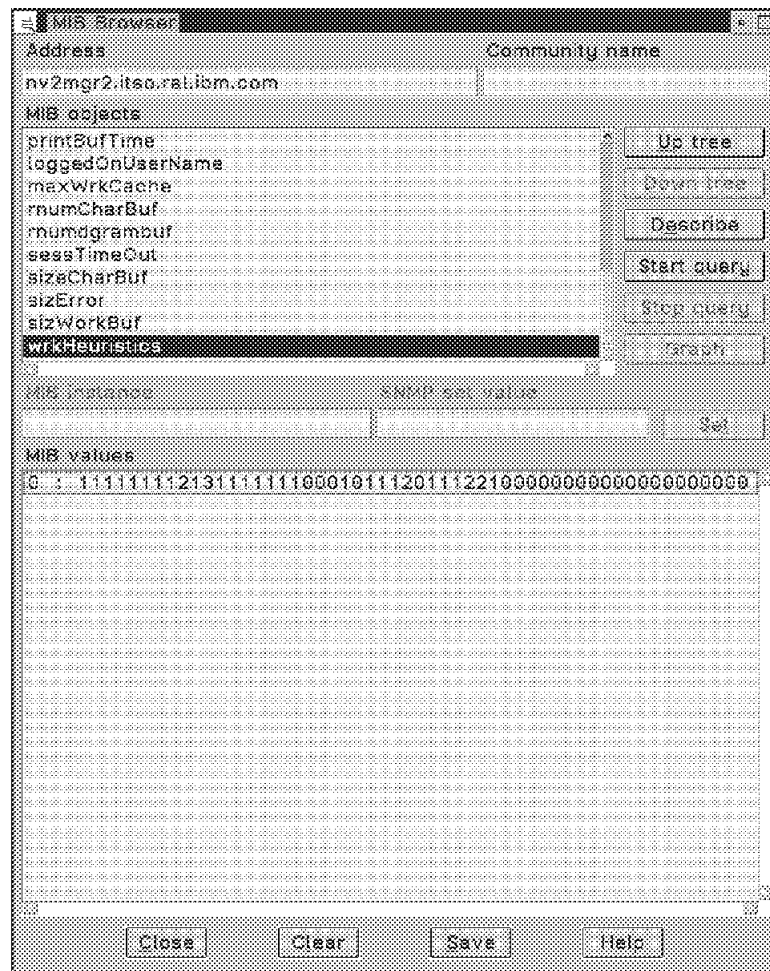


Figure 136. Work Heuristics

5.2.3 Statistics

The *IrStatistics* group contains information on items that affect the performance of LAN Requester. These variables are read-only, but the *Command* group variable *clearStatistics*, can be used to reset the information. These variables might be candidates for a MIB application that monitors thresholds and then creates alerts.

The next example shows the rBigBufNfeed variable. The information displayed reflects the number of times the Requester tried but failed to allocate a big buffer.

The following is the tree structure to get to the MIB variables shown:

```
* private
  enterprises
    ibm
      ibmProd
        os2LReq
          lrStatistics
```

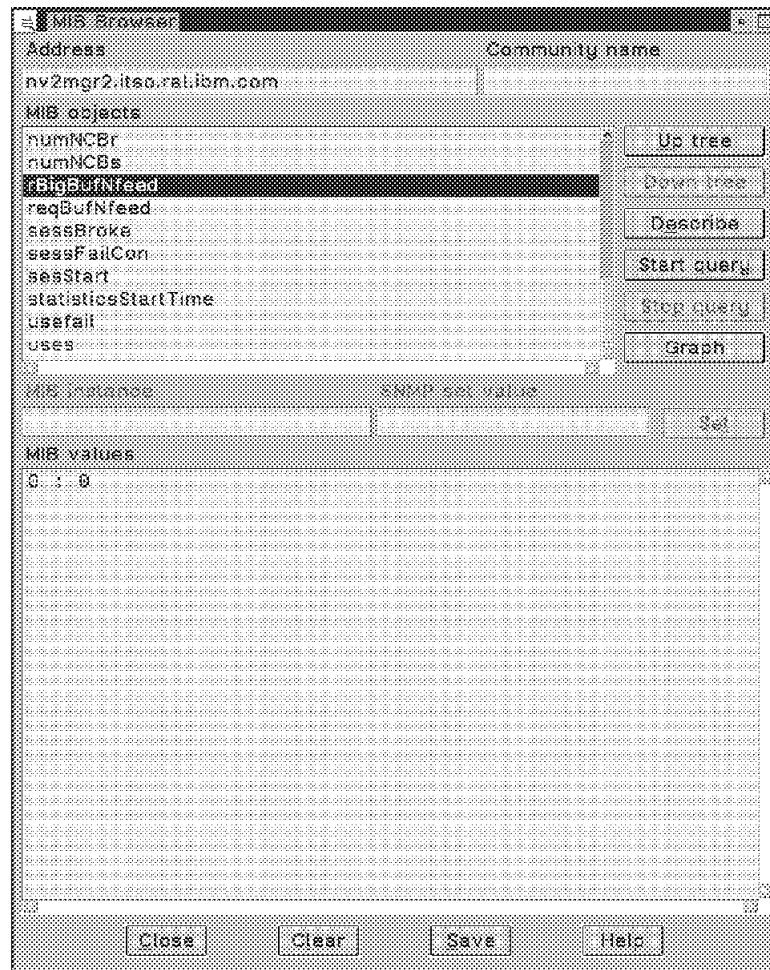



Figure 137. Failed Requester Big Buffer Allocation

5.2.4 Commands

The *Command* group allows you to activate, deactivate, or pause LAN Requester service, and well as to clear statistics. The next example shows the variables in the *Command* tree.

The following is the tree structure to get to the MIB variables shown:

```
* private
  enterprises
    ibm
      ibmProd
        os2LReq
          Command
```

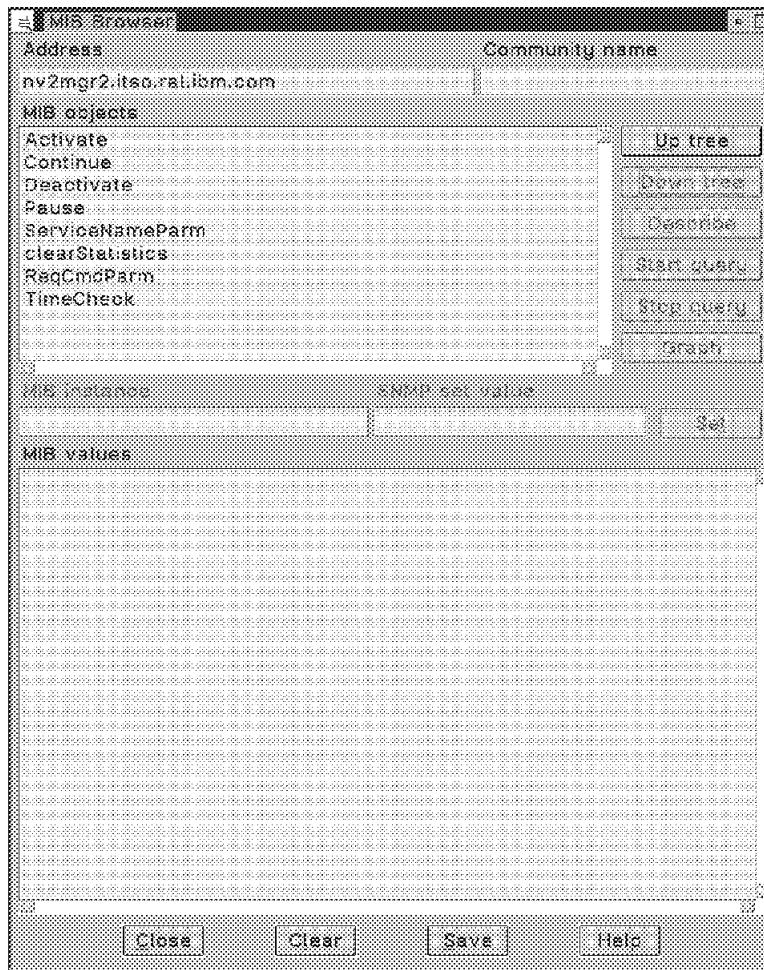


Figure 138. Command Options

5.2.5 Performance

When you start the LAN Requester subagent, 25 performance counters are also started. The next example shows the *IrPollPerf* variable. This variable defines the polling interval for collection of the performance metrics. This is a read/write variable and SETing it to a value of 0 disables the polling and stops the collection of information.

The following is the tree structure to get to the MIB variable shown:

```
* private
  enterprises
    ibm
      ibmProd
        os2LReq
          IrPerfMetrics
```

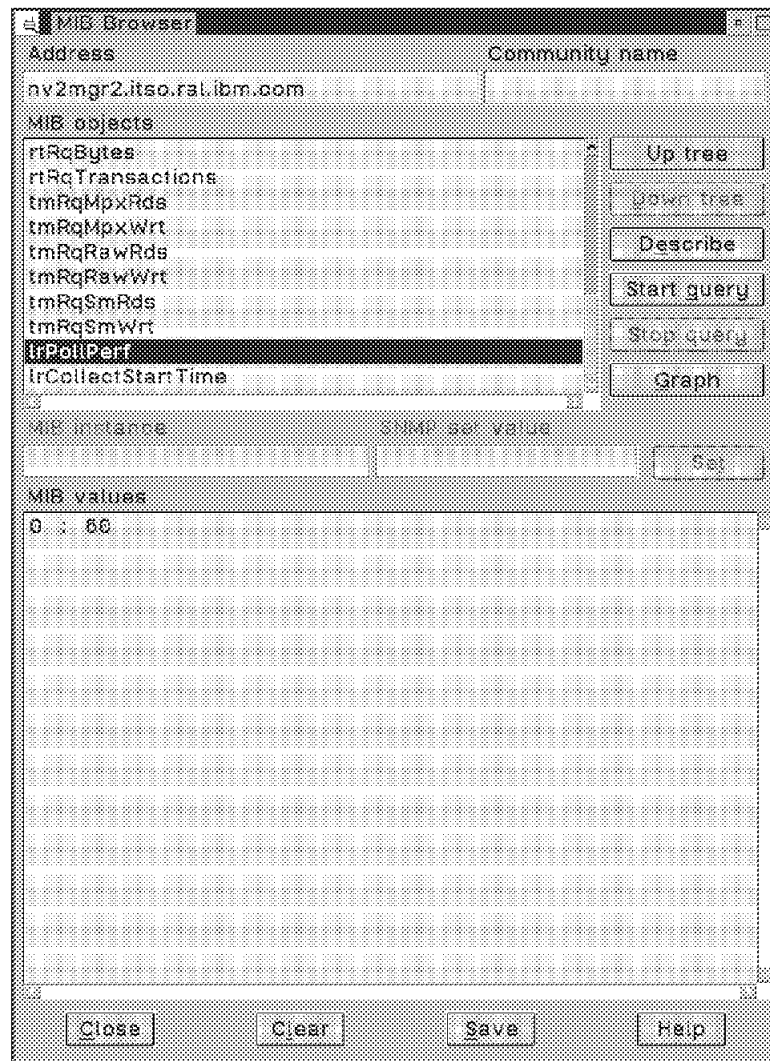


Figure 139. LAN Requester Poll Performance

Chapter 6. DOS and DOS/Windows Agents

NetView for OS/2 does not provide SNMP agents for DOS or DOS/Windows workstations. Management of these workstations is accomplished through the use of LMU agent code together with the LMU-supplied proxy agent. LMU alerts are converted to traps by the proxy agent and are forward to NetView for OS/2. Information on LMU topology, configuration, and performance can be accessed by browsing the LMU MIB on the proxy agent workstation.

6.1 DOS Agents

The LMU DOS agents provide two basic management functions:

- Initial heartbeat to the managing system

There is no polling of DOS workstations by LMU so it is up to the user to notify the managing system that they have joined the LAN. This can be achieved by including the heartbeat command (LMUDOSHB.COM) in the user's AUTOEXEC.BAT, or by invoking the command from any other DOS procedure.

- Collection of configuration information

LMU provides a command (QDOSVPD.COM) that obtains Vital Product Data (VPD) as well as monitors critical system files. This information can be collected and sent to the LMU database on the LMU managing station. This information is only collected if the user sends it. There is no way for NetView for OS/2 to directly request it. Typically, this command would be included in the AUTOEXEC.BAT file of the workstation.

The following is an example of the AUTOEXEC.BAT on a DOS workstation that is managed by LMU. At system boot up, a heartbeat will be sent to LMU managing system LMUMGR, and VPD information will be collected and sent to the LMU database on LMUMGR.

```
@ECHO OFF
PROMPT $p$g
SET PATH=C:\DOS;C:\;
LMUDOSHB LMUMGR
QDOSVPD /r LMUMGR
```

The VPD information, as stored in the LMU database, can be viewed by browsing the LMU MIB of the proxy agent on the LMU managing workstation with the MIB Browser, or by using the Query Manager function of DB2/2. The following two examples show the respective views.

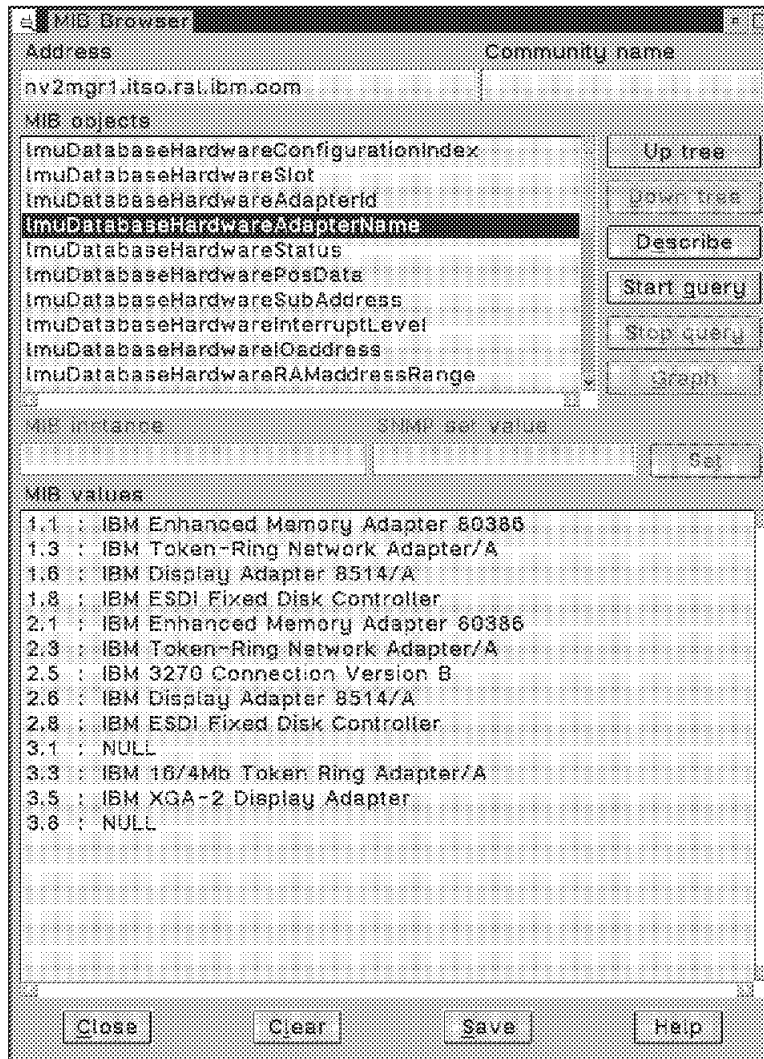


Figure 140. MIB Browser View of LMU Database

Report			
COMPUTER	SLOT	ADAPTER ID	ADAPTER NAME
WTR32216	1	EDDF	IBM Enhanced Memory Adapter 60386
00000009:40003003372	1	8EFC	-
00000009:40003003372	3	E001	IBM 16/4Mb Token Ring Adapter/A
00000009:40003003372	5	8FDA	IBM XGA-2 Display Adapter
00000009:10005A31F459	1	EDDF	IBM Enhanced Memory Adapter 60386
NVSRV30	1	8EFF	IBM PS/2 SCSI Adapter with Cache
WTR32216	3	E000	IBM Token-Ring Network Adapter/A
WTR32216	6	8F7F	IBM Display Adapter 8514/A
WTR32216	8	EDFF	IBM ESDI Fixed Disk Controller
00000009:40003003372	9	8FA4	-
00000009:10005A31F459	3	E000	IBM Token-Ring Network Adapter/A
00000009:10005A31F459	5	E1EF	IBM 3270 Connection Version B
00000009:10005A31F459	5	8F7F	IBM Display Adapter 8514/A
00000009:10005A31F459	9	EDFF	IBM ESDI Fixed Disk Controller
NVSRV30	3	E001	IBM 16/4Mb Token Ring Adapter/A
NVSRV30	5	8FEB	IBM XGA Video Adapter

Figure 141. Query Manager View of LMU Database

6.2 DOS/Windows Agent

The LMU Windows agent provides the following systems management functions:

- Periodic heartbeats to the managing system
- Collection of configuration information
- The ability to execute DOS commands and run DOS applications from the managing system

The process for collecting data and the information that is available is the same as for the LMU DOS agent.

The following examples show how a command can be issued to the Windows workstation and how the output is displayed.

To get access to the LMU remote command function, you must first select the object and use the right mouse button to get the Options menu. From this menu you select **Application action**, which will allow you to select the LMU Remote Command option.

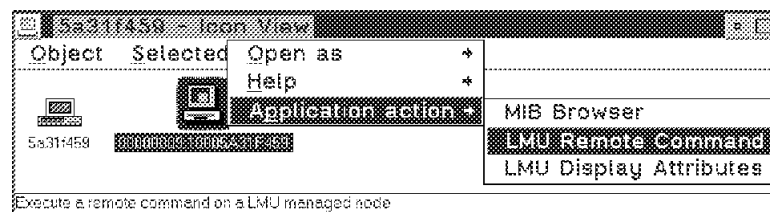


Figure 142. LMU DOS/Windows Object

This action will open the LMU GUI - Execute Commands window. You can enter DOS commands or an application name using the Command field. In this case the QUERYVPD command was issued without the output being sent to the database. Instead it was returned to the command window.

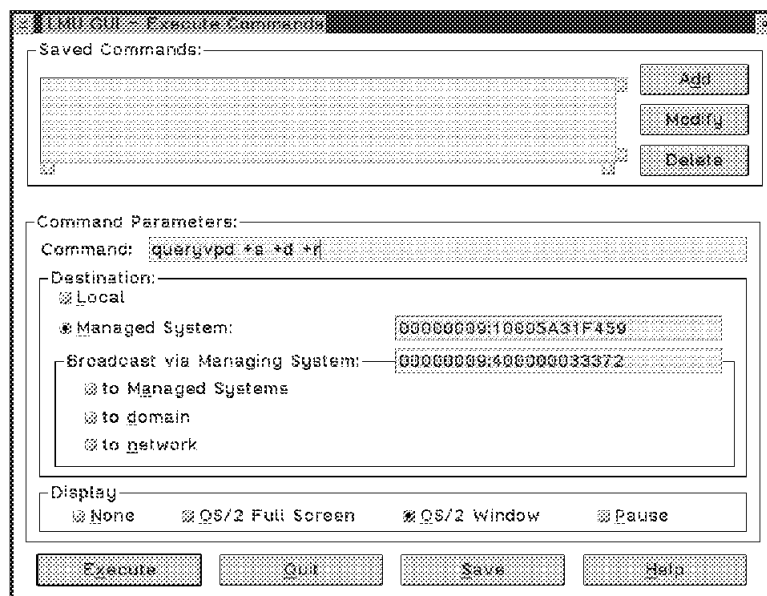
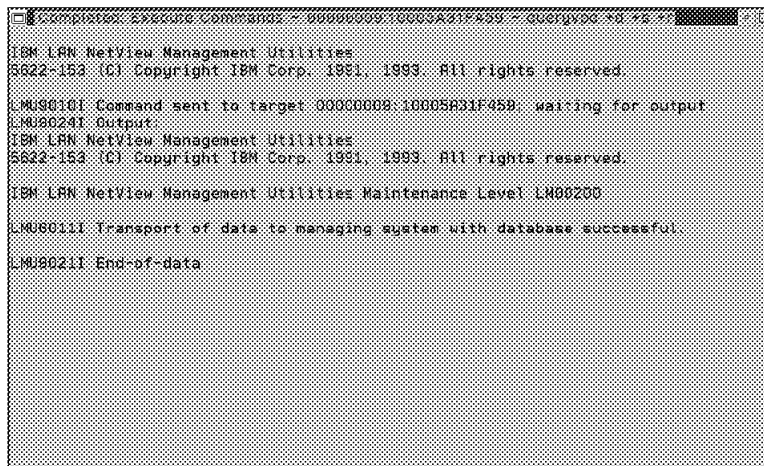


Figure 143. LMU Remote Command - Execute Panel

When the command is executed, a Results window is opened. Here you can see that the command completed successfully and the data was sent to the LMU database.



```
Computer: Execute Command: 00000000:1000A21F459 / Q0000000 *d *e *f
IBM LAN NetView Management Utilities
S622-153 (C) Copyright IBM Corp. 1991, 1993. All rights reserved.

LMU9010I Command sent to target 00000000:1000A21F459: waiting for output
LMU9024I Output:
IBM LAN NetView Management Utilities
S622-153 (C) Copyright IBM Corp. 1991, 1993. All rights reserved.

IBM LAN NetView Management Utilities Maintenance Level LM00200
LMU9011I Transport of data to managing system with database successful.
LMU9021I End-of-data
```

Figure 144. LMU Remote Command - Results Panel

Chapter 7. NetWare Agents

NetView for OS/2 manages NetWare servers using agent code and management applications supplied as part of LMU. Any alerts produced by the NetWare server will be converted to SNMP traps by the proxy agent and forwarded to NetView for OS/2.

7.1 LMU Agents for NetWare

The following management functions are available:

- Execution of management applications, through commands issued using the LMU remote command facility

LMU provides NetWare Loadable Modules (NLM) that collect VPD data (QUERYVPD.NLM), monitor performance of the server (NSVWATCH.NLM) send alerts to the management system when selected conditions occur (SS.NLM), and collect volume data (VOLWATCH.NLM) from the server disks. A SHUTDOWN.NLM is also provided that allows the shut down and, optionally, restart of the server. NSVWATCH requires that the NetWare module SS.NLM be installed on the server.

- Use of the LMU remote command facility to execute commands on the server console, provided that they don't require a user response

There is no full-screen interface to the server console, so any command issued will get an immediate Completed in the command window along with the message:

LMU9023W Console output from target is not supported

This message is normal and will be issued for any command issued to the server.

- Issue of periodic heartbeats to the managing system

7.2 Management Desk - LMU NetWare Objects

The NetView for OS/2 Management Desk provides an application that allows the attributes of an LMU-managed workstation to be displayed. The following examples show three different types of NetWare objects, along with their associated attribute information.

When you select an LMU-managed workstation, the LMU Display Attributes function is accessed using the Application action menu.

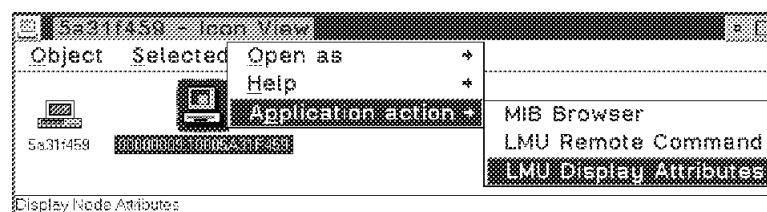


Figure 145. LMU NetWare Object

In this case, we have selected a Windows workstation that is running NetWare Requester. The LMU Display Attributes were:

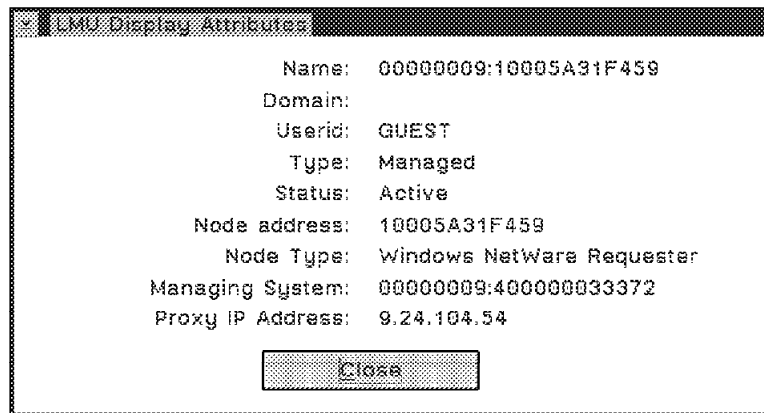


Figure 146. LMU Display Panel - NetWare Requester

The information that was reported is the name of the machine split up into the network address and the systems MAC address, login name, IPX address of the proxy system and other useful information. This can be useful if you wish to know if the system is an OS/2, DOS or Windows system.

The next example shows the LMU icon for a NetWare server.

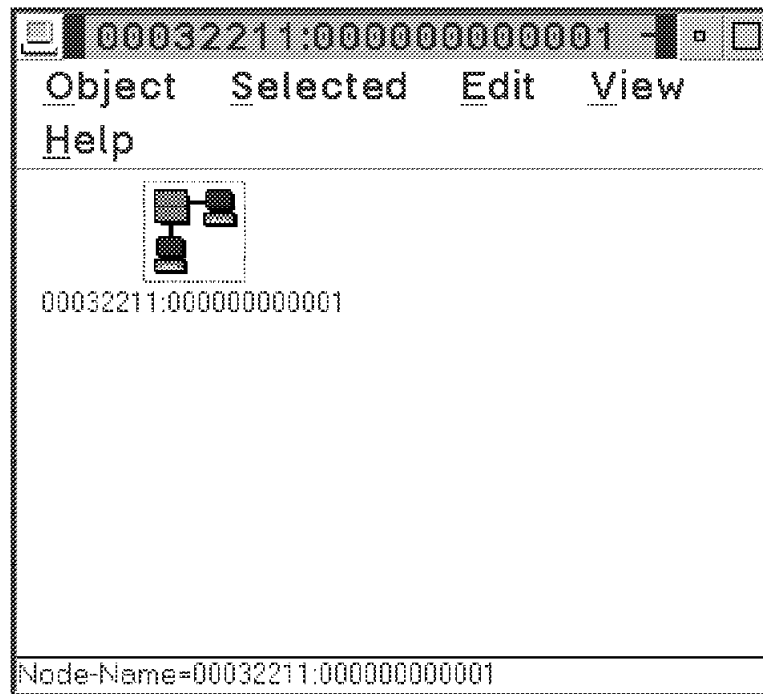


Figure 147. LMU Icon - NetWare Server

Figure 148 on page 151 shows the LMU Display Attributes panel for this workstation.

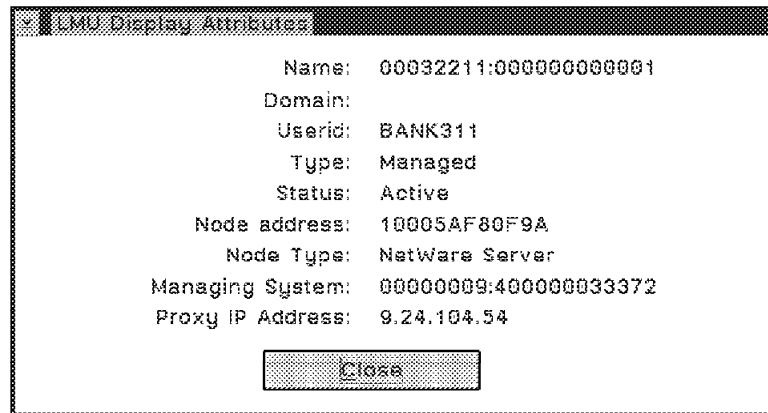


Figure 148. LMU Display Panel - NetWare Server

Finally, we will select an LMU-managed OS/2 workstation that also has NetWare Requester installed. Notice that there is one icon for:

- IP host name
- LMU IPX address
- NetView for OS/2 discovered IPX address

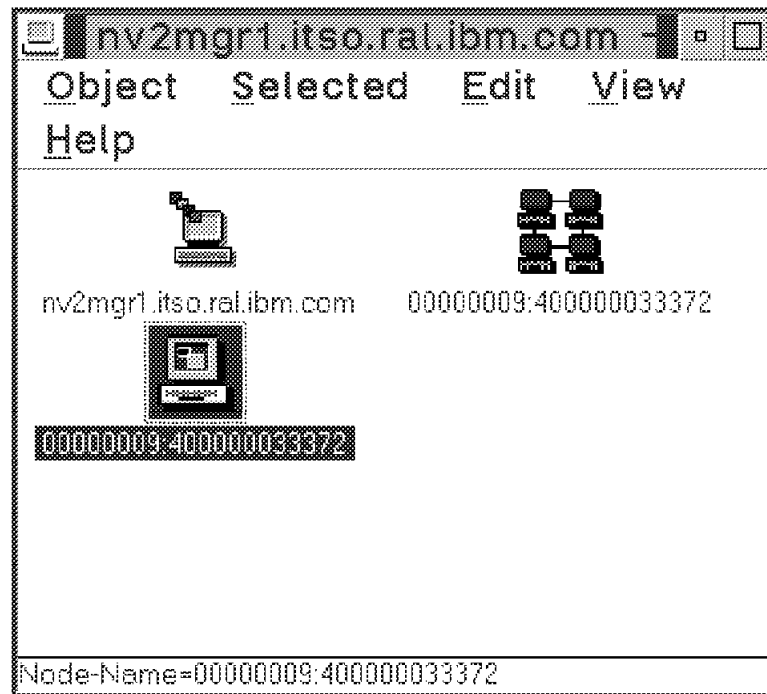


Figure 149. All the Different Icons for Our NV2MGR1 Machine

The LMU Display Attributes panel confirms the information about the workstation. The node type of *OS/2 Interoperable* is the standard description for OS/2 NetWare workstations as shown in Figure 150 on page 152.

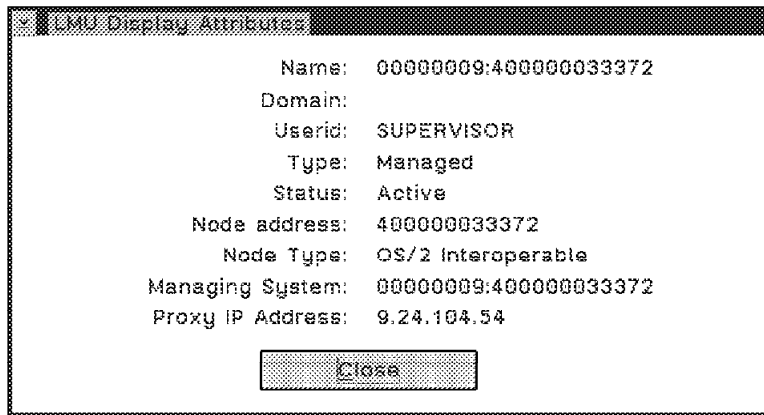


Figure 150. LMU Display Attributes - Showing OS/2 Interoperability

7.3 Management Desk - NetWare Objects

The Management Desk also provides some applications for NetWare objects discovered using NetView for OS/2, rather than LMU. These applications provide information on node status as well as node setup. The following examples show how the applications are selected and what information is displayed.

When you select a NetWare object two functions are available using the Application action menu. They are Quick status and Get node information.

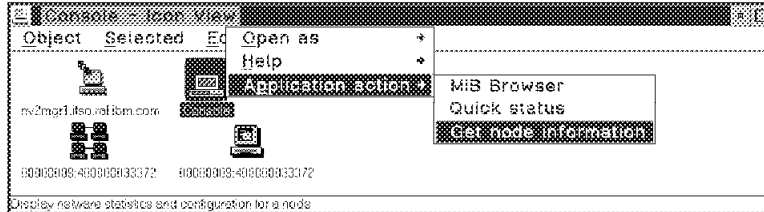


Figure 151. NetWare Object

The next example shows the result of using **Quick status**:

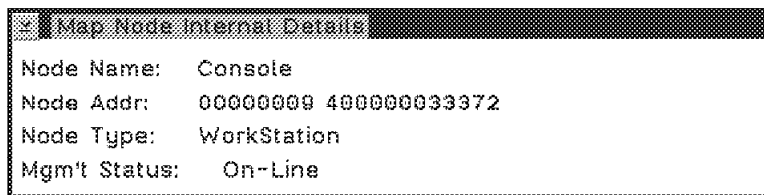


Figure 152. NetWare Quick Status - Results

Finally, the next two examples show the panels displayed using the Get node information function. It is a multi-page panel. The first page shows which version of NetWare is running so that all drivers can be kept up to date, or in case you are running old versions of the IPX stack for NetWare 3.11 you can identify systems to upgrade to NetWare 3.12. Of course, the same would hold true for NetWare 4.01.

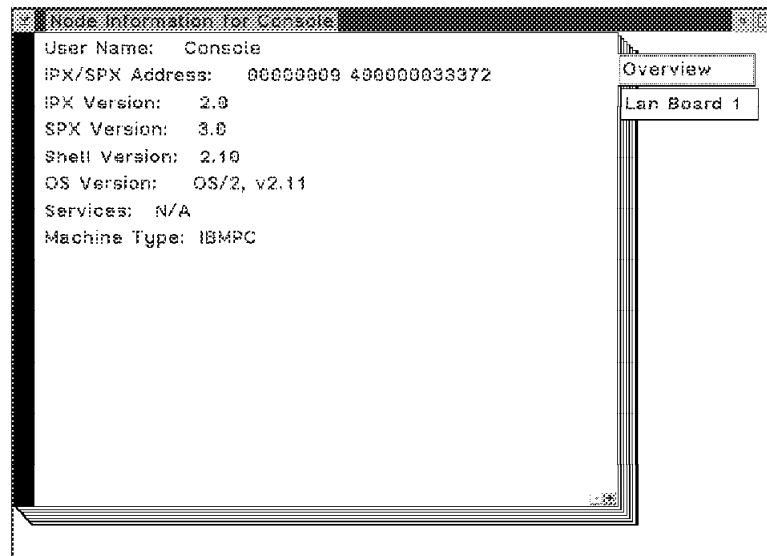


Figure 153. NetWare Node Information - Overview

In large networks where performance tuning is an issue the **LAN Board 1** page provides statistical information on the network card in the system which can be used in performance tuning or in identification of failing cards.

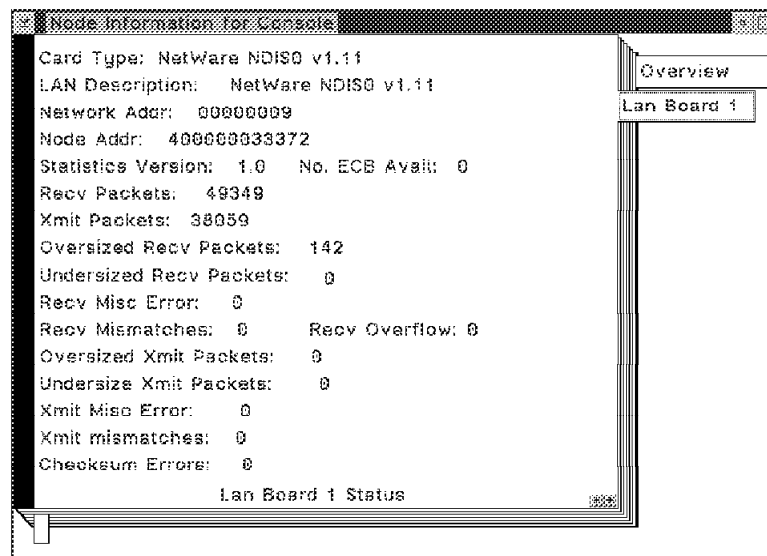


Figure 154. NetWare Node Information - LAN Boards

Chapter 8. LMU Interactions

LMU is an application that has been integrated into NetView for OS/2. We will show how to install and configure LMU in a NetView for OS/2 environment. We will also show how to set up an LMU managing station to be a proxy agent for a centrally located NetView for OS/2 Manager of Managers. With this capability, NetView for OS/2 can manage stations with the NetView for OS/2 agent code, and all LMU managing stations and the stations they manage.

Finally, we will show some of the management utilities that LMU offers and how they can be integrated into a NetView for OS/2 environment.

8.1 LMU Installation and Customization

The LMU installation process is dependent on the platform that it will be executing on. In this section, we will show the steps used to install the LMU components for OS/2, DOS, Windows and for a NetWare server.

Remember that the last three components can only be managed stations and that the LMU code needs to be installed in at least one OS/2 machine to act as the managing station. This terminology is explained better in 8.2, “LMU Configuration” on page 157.

8.1.1 LMU Installation for OS/2 Stations

The installation utility is called LMUINST. It is located on LMU diskette 1. If this utility is called with no parameters, it will show the parameters available, just like a help command would. We started the installation process with the options:

```
a:\lmuint all /Td
```

Where:

- all specifies that all the LMU programs and files (for all platforms), the graphical user interface and the documentation are to be installed from the LMU diskettes.
- /Td specifies that the target drive is d, instead of the boot drive.

This target drive may be either a local or network drive.

You must customize the LMU software prior to using it. Refer to 8.2.1, “LMU Configuration for OS/2 Stations” on page 157 for details on this customization.

8.1.2 LMU Installation for DOS Stations

If the LMU software was previously installed on a shared disk to which the DOS station has access, no additional action needs to be performed.

If that is not the case, then copy the following files from the LMU diskette 4 to the DOS machine:

- DOSVIRGA.COM
- QDOSVPD.COM
- LMUDOSHB.COM
- AUEDOSAL.COM
- ADAPTERS.TBL
- CVT_VPD.EXE

- USERVPD.SMP
- ADAPTERS.SMP
- CRITFILE.SMP

The usage of these files is explained in 8.2.2, “LMU Configuration for DOS Stations” on page 164.

8.1.3 LMU Installation for Windows Stations

To install the Windows portion of the LMU code, use the LMUINSTW utility located on diskette 4 as follows:

```
a:\lmuinstw z:
```

where z: is the target drive.

This utility creates the LMU2 directory on the specified drive and copies the necessary files to it.

8.2.3, “LMU Configuration for Windows Stations” on page 164 shows how to customize this environment.

8.1.4 LMU Installation for NetWare Servers

If your NetWare server is going to maintain the LMU code for both itself and its requesters, use the installation utility LMUINST on the LMU diskette 1, as follows:

```
A:\lmuinstant network /Tg
```

Where /Tg is the target drive g. Make sure that you are specifying a shared drive that belongs to your NetWare server.

If you are going to install only the server code, copy the following files from the LMU diskette 4 to a server’s partition:

- All .NLM files (*.NLM)
- QDOSVPD.COM
- ADAPTERS.SMP
- CRITFILE.SMP
- LMUBIND.SMP
- USERVPD.SMP
- ADAPTERS.TBL

This is done at a NetWare requester and the server partition is one of the shared drives that was mapped at this station.

Also, copy the following files to the server’s DOS partition:

- LMUDOSHB.COM
- CVT_VPD.EXE

To access the DOS partition, either access it before starting the NetWare server or after bringing it down.

The customization of the NetWare server is shown in 8.2.4, “LMU Configuration for NetWare Servers” on page 165.

8.2 LMU Configuration

When configuring LMU, one of the tasks is to decide what role each machine will play in the LMU environment.

In general terms, a machine can be:

- A managing station: it is a NetWare requester or OS/2 LAN Server requester that receives configuration and performance information from the managed workstations and stores the data in an OS/2 database. It also directs alerts to the fault management system and monitors the heartbeat status of managed workstations.
- A managed station: it is a node in the NetWare or OS/2 LAN Server environment (server or requester) that:
 - Sends configuration information to a managing system for inclusion in an OS/2 database (any platform).
 - Directs alerts to the fault management system (any platform).
 - Executes remote commands received from an administrator workstation that do not require a user response (OS/2, Windows, NetWare Server or Macintosh based).
 - Issues periodic heartbeat signals to the managing station to indicate that the station is alive (OS/2, Windows, NetWare Server or Macintosh based).
- Administrator workstation - A NetWare requester or OS/2 LAN Server requester logged on as an administrator/supervisor that has the ability to direct remote commands to a managed station.
- Fault manager - A requester (NetWare or OS/2 LAN Server) that receives alerts from managed stations, performs some action based upon the alert value, and forwards this alert to IBM NetView or IBM LAN Network Manager.
- Proxy agent station - A NetWare requester or OS/2 LAN Server requester that contains an SNMP agent, called LMUSNMPD, that converts LMU alerts to SNMP traps. It then sends data to the NetView for OS/2 program which can send commands back. Examples of this interaction are shown in 8.5, "Integrating LMU Alerts into NetView for OS/2" on page 169.

A single OS/2 workstation can perform more than one of these defined actions. For example, in our environment we have configured one machine to be a managing, managed, fault manager, proxy agent and administrator machine. In addition, it maintains the LMU database running DB2/2.

8.2.1 LMU Configuration for OS/2 Stations

This section covers the configuration of the OS/2 station that has just been described. The key points are the customization of the `lmu.ct1` file described in "General Customization File" on page 160 and the usage of the LMUCUST utility, explained in more detail in 8.2.1.3, "Preparing the Environment" on page 162.

8.2.1.1 IBM TCP/IP for OS/2 Configuration

Refer to 2.5.2, "Installing TCP/IP" on page 23 for a description of how to configure the base IBM TCP/IP for OS/2.

You have to update the MIB file to include the LMU-related MIB. To do that, execute the following command:

```
copy \tcip\etc\mib2.tbl + \lm2\lmumib.tbl = \tcip\etc\mib2.tbl
```

The administrator workstation has to be enabled to receive the remote commands that will be forwarded to the target managed stations and executed there. The configuration consists of defining a user ID and a password for the REXEC command, as shown in Figure 155. This is done through the TCP/IP Configuration application (tcpipcfg.exe) located inside the TCP/IP folder.

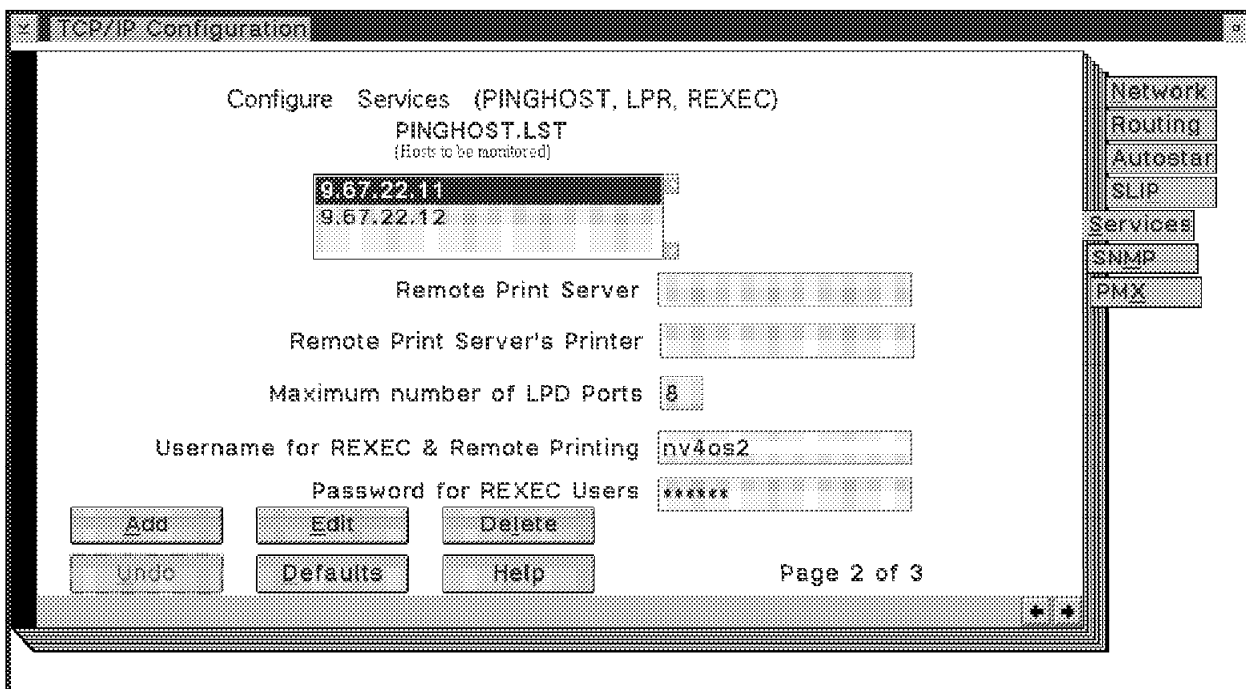


Figure 155. IBM TCP/IP for OS/2 Configuration - REXEC Customization Panel

Notice that the password is not shown on the above panel for security reasons; however, the following two lines are added to your config.sys file:

```
SET USER=NV4OS2  
SET PASSWD=xxxxxx
```

NV4OS2 is what we used for the user ID.

Note: When you try to execute a command using LMUCMD on a remote workstation, it will prompt you for this Username and Password. Sometimes LMUCMD will be executed from within a program and there will be no user interface. There is a way for you to pass this security information along with the request. See 8.4, "LMU Remote Commands from NetView for OS/2" on page 168 for a description of what is required to do this.

Do not forget to specify that you want the REXEC daemon to be started automatically. This daemon is required for remotely executing LMU commands

through the proxy agent. How to configure TCP/IP to automatically start the daemon is shown in Figure 156 on page 159.

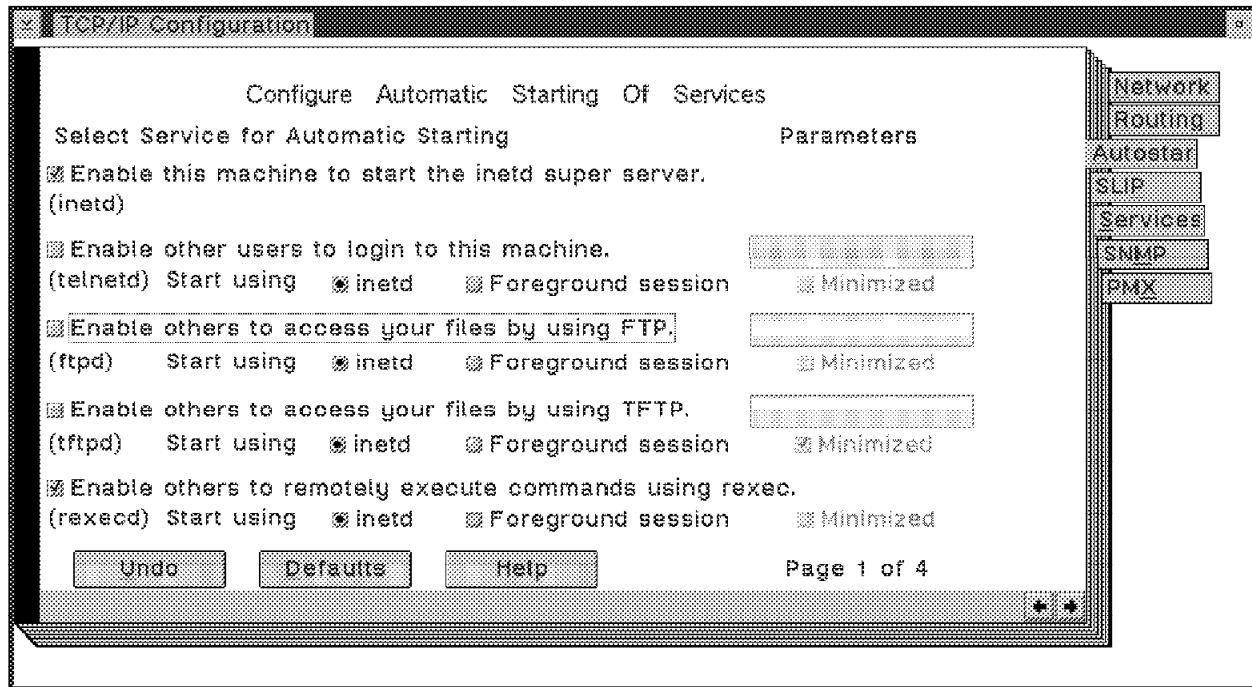


Figure 156. IBM TCP/IP for OS/2 TCP/IP Configuration - Autostar Panel

Important

If you want the alerts/traps sent from the LMU proxy agent (an LMU managing station) to go to a centralized NetView for OS/2 manager, you must configure the SNMP trap destination file to point to the address of this NetView for OS/2 managing station. This is shown in Figure 157 on page 160.

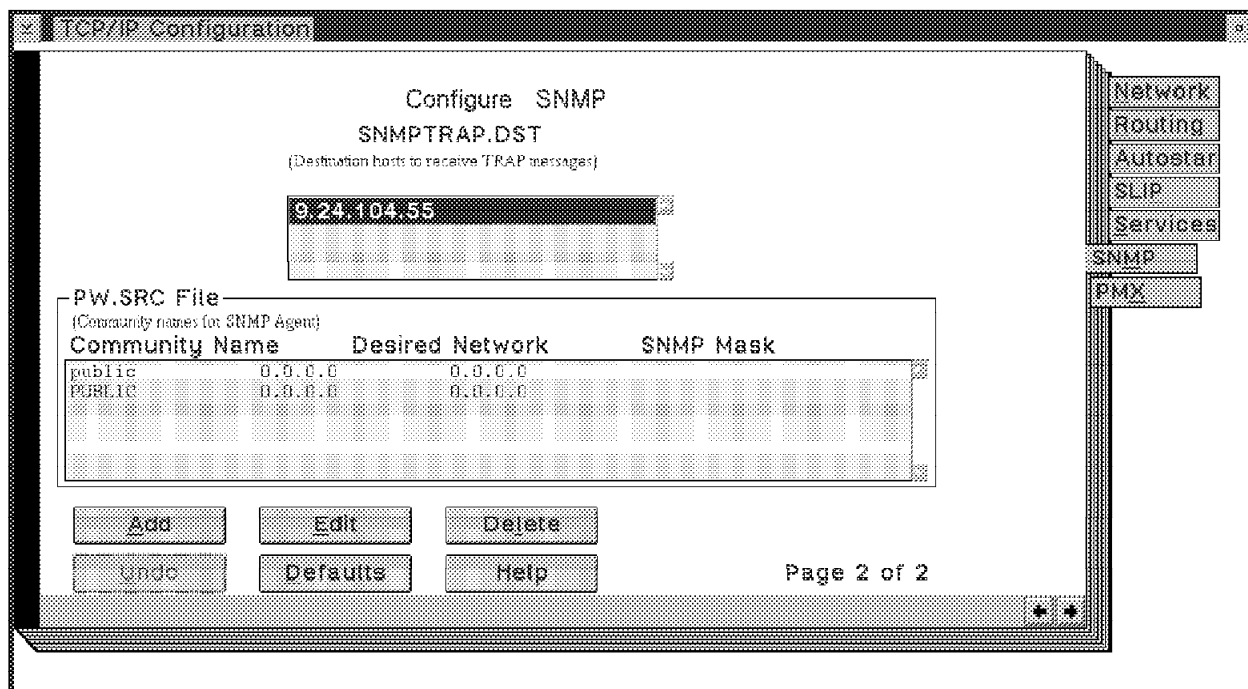


Figure 157. IBM TCP/IP for OS/2 Configuration - SNMP TRAPDST Configuration

In the above panel, we added the TCP/IP address of the NetView for OS/2 managing station which was also the Manager of Managers (MOM). The MOM's address was **9.24.104.55**.

Note: Even if the LMU proxy agent and the NetView for OS/2 MOM are on the same machine, you must still update this field to point to its own IP Address.

8.2.1.2 LMU Control Files Configuration

There is an LMU program called LMUCUST that automates most of the actions required to customize a workstation, but there is some post-customization that will need to be done manually.

All of the configuration files are in the LMU2 directory.

General Customization File: The LMUCUST program reads the `lmu.ct1` file in order to perform the customization. There is a sample file called `lmuct1.smp` in the `\LMU2` directory that should be used as a base. To create your own `lmu.ct1`, type the following command:

```
copy lmu2\lmuct1.smp lmu2\lmu.ct1
```

This file specifies where to find the managing stations, the proxy agent, the fault manager and other components. It does not define the role of this machine. When you execute the LMUCUST program using the `lmu.ct1` file, it sets up the LMU environment.

The changes we have made were:

- Substituted all *computername* or *internetwork address* references in the ASCIIZ entries to be 00000009:400000033372. This number is the ID of our machine in the NetWare network and consists of the NetWare network number and the MAC address of the machine. If you are in an OS/2 LAN Server environment, you would change it to the value of the

COMPUTERNAME field that is in the IBMLAN.INI file located in the IBMLAN directory.

- Modified all the drive references from C: to D:, since we have installed the LMU code on the D drive.

The only exception was the statement:

```
DEFINE_PROFILE INI_FILE(C:LMU.INI)
```

because this file needs to reside on the OS/2 boot drive and must have the name LMU.INI.

- When defining the proxy agent to be accessed, you also have to specify its name:

```
APP(LMU_UTILITY),  
    KEY(SNMP_PROXY_AGENT),  
    ASCIIIZ(1musnmpd,00000009:400000033372);
```

The sample file also has the characters '...' indicating that you may address more than one proxy agent. Whether you have one or many proxy agents, you need to remove these characters from the statements.

There is a copy of the lmu.ct1 file that we have used in A.1, "LMU/2 Control File (LMU.CTL) for the Managing System" on page 275.

User-Provided Configuration File: Another file to customize is the uservpd.dat, which provides user-chosen information to the QUERYVPD program. This configuration data will be sent to the LMU database during the customization process.

There is sample file named uservpd.smp that can be edited and used as a base file. Copy this file to uservpd.cfg and run the LMU CVT_VPD utility against it to produce the uservpd.dat, as follows:

```
cvt_vpd uservpd.cfg uservpd.dat
```

The uservpd.cfg file that we have used can be found in A.2, "USERVPD.CFG File" on page 282.

Fault Manager-Related File: There is a sample file called aueuser.smp that can be used to set up the generic alert table that is used by the fault manager. It provides some sample definitions that can be easily modified. We copied it to a file called aueuser.tab A sample entry in this file is:

# Product/ # Appl Id	Alert Type	Alert Desc	Source	Target	Auto Thresh	Notify Thresh	Auto Timer	Notify Timer	Command
# ----- 5622153	11	C000	*	*	0000Y	0000Y	0	0	-\$B Alert from -\$c is -\$t

The sample table comes with the parameter -\$g in the command field. This value indicates that the alerts received by this machine will be sent to the LMU graphical user interface. You might want to change this value to -\$B in all entries, so that it will send the alerts both to the LMU GUI and the LMU SNMP proxy agent.

You can run the AUEVERTB utility using this table to check for syntax errors. This utility assumes that the lmu.ini exists on the same drive as the OS/2 operating system (usually, the C drive). This utility needs the lmu.ini file because it sends its output to the log file defined there. You will only be able to

use this utility after you have run LMUCUST, since it is LMUCUST that creates the lmu.ini file.

As this table is read-only at the startup of AUERECVR (the process that actually performs actions against the alerts received), whenever a change is made to this file, the task will need to be restarted. To restart the task, enter:

```
lmuquery /tf auerecvr  
detach auerecvr
```

8.2.1.3 Preparing the Environment

After updating the ctlfile, we were able to use the LMUCUST program to customize LMU. The command we used was:

```
LMUCUST managing alerts administrator lan_manager fault_reporter  
managed proxy_db /Td
```

You will need to execute this command from the \LMU2 directory or specify the directory to be inserted in the OS/2 config.sys file. The \LMU2 directory will be automatically inserted in the config.sys when LMUCUST is run.

The options that we used were:

- managing - The station will be a managing system, listening for heartbeats and maintaining the LMU database.
- alerts - The managing station will generate alerts and send them to either NetView or LAN Manager.
- administrator - The station will be able to issue controlling commands.
- fault_manager - The station will handle alerts that originated from another workstation.
- lan_manager - The station will forward alerts to the LAN management functional address. In our case, this will be the NetView for OS/2 managing station, which can forward alerts to NetView on the host using the Host Connection program.
- fault_reporter - The station will route generic alerts from OS/2 subsystems and applications to the Fault_Manager.
- managed - The station will receive controlling commands and send requested information to the managing system.
- proxy_db - The workstation will start the LMU proxy agent (LMUSNMPD) and access the LMU database when requests for information are issued.
- /Td - The target drive for storing the configuration information is the d drive.

The output of this command is shown in Figure 158 on page 163.

If you want help on the LMUCUST options, enter LMUCUST ?.

If you are configuring a managed station only, you should run the LMUCUST program with the following parameters:

```
LMUCUST managed /Td
```

IBM LAN NetView Management Utilities
5622-153 (C) Copyright IBM Corp. 1991, 1993. All rights reserved.

IBM LAN NetView Management Utilities Maintenance Level LM00200

LMUCUST: If you have not already created a fault manager alerts table,
do so, based on AUEUSER.SMP.

LMUCUST: LMU.INI built from D:\LMU2\LMU.CTL.

LMUCUST: USERVPD.SMP is in your D:\LMU2 subdirectory.
See the IBM LMU/2 User's Guide for more information concerning
modifying USERVPD.SMP to make a USERVPD.DAT file.

LMUCUST: Cannot locate line in STARTUP.CMD that starts the
server or requester. You must manually add
'CALL D:\LMU2\LMUSTART.CMD' to your STARTUP.CMD.

LMUCUST: LMUCUST does not increase the number of NETBIOS resources available
to LMU. Modify the NETBIOS resources as needed.

LMUCUST: Customization complete, restart the computer to activate changes.

Figure 158. LMUCUST Command Output

Running the LMUCUST utility results in the LMUSTART.CMD file being created, the CONFIG.SYS file being updated, the LMU.INI file being created and the LMU folder being created on your workstation. LMUCUST will also try to update your STARTUP.CMD file so that LMU will be automatically started when the machine reboots. Since we had no entry in the STARTUP.CMD file to start the Requester service, the LMUSTART.CMD command was not inserted.

Note: If your proxy agent is not running in the station defined by the environment variable HOSTNAME in config.sys or the community name to access the MIB variables in that machine is not public, you will have to specify these values in the proxy agent definition, that is, in the lmustart.cmd file. Chapter 8.3, "LMU Startup" on page 166 has an example of this entry. It is necessary to restart the OS/2 system to make these changes effective.

Following is a copy of the LMUSTART.CMD that was created as a result of running our LMUCUST routine with all our parameters:

```
@ECHO OFF
REM D:\LMU2\LMUCUST.EXE Maintenance Level: LM00210
REM LMUCUST parameters: /TD MANAGING ALERTS ADMINISTRATOR FAULT_MANAGER LAN_MANAGER
    FAULT_REPORTER MANAGED PROXY_DB
DETACH D:\LMU2\AUECATCH.EXE 2
DETACH D:\LMU2\AUERECVR.EXE
START "LMU Managing System" /C /MIN D:\LMU2\LMUSRV.EXE /A
D:\LMU2\LMUSLEEP.EXE 120
START D:\LMU2\LMUCLI.EXE
START "LMU SNMP Proxy Agent" /C D:\LMU2\LMUSNMPD.EXE /D
```

Figure 159. LMUSTART.CMD File

8.2.2 LMU Configuration for DOS Stations

There is not much to customize in the DOS environment. The minimum function that needs to be configured is to issue a heartbeat when the machine is started. This is done by inserting the following line in the `autoexec.bat` file:

```
C:\mu2\mudoshb.com managing_system
```

The value *managing_system* can be either a computer name (if in the OS/2 LAN Server environment) or an internetwork address (if it is in a NetWare environment).

This line could also be included in the `profile.bat` file or in the NetWare login script.

Use the parameter `?` to display the syntax for this command.

The DOSVIRGA.COM program sends an alert to a fault manager station warning that a virus is detected. It does not detect the virus itself. You will have to run a virus-detection program, check the return code from this program and if the "virus detected" condition is satisfied, call DOSVIRGA. It will send an alert to the fault manager defined by the SET FAULT_MANAGER statement that you define in the `autoexec.bat` file.

If you want to store specific information about your machine in the OS/2 database, customize the `uservpd.smp` in the same way that it was done in "User-Provided Configuration File" on page 161. Use the QDOSVPD.COM program which is the equivalent of OS/2 QUERYVPD in the DOS environment to send the information to the database. To execute it, type: `QUERYVPD /R destination` where *destination* is either the computer name (from the IBMLAN.INI file) or the internetwork address.

The AUEDOSAL.COM program generates alerts if you are going to do your own alert generation.

8.2.3 LMU Configuration for Windows Stations

Before proceeding to the actual configuration, make the LMU2 directory (or the one in which the Windows portion of the code was installed) the current directory.

We used the following procedure to customize the Windows workstation:

- If you have already configured the `lmu.ct1` file as described in "General Customization File" on page 160, copy it to the LMU2 directory. The configuration utility will use this file to customize the `lmu.ini` file that is used in the Windows environment to specify the LMU-related variables.

If you have not customized the `lmu.ct1` file, then copy the `lmuiniw.smp` file to `lmuw.ini` and fill in the appropriate information:

- `managing system` - This is the station that will receive the heartbeats.
- `managing_system_with_database` - This is the station that will maintain the LMU database.
- `fault_manager` - This is the station that will receive the alerts that are generated by the workstation.

- pulse_rate - This is the frequency of the heartbeat. A value of zero means only the initial and terminating heartbeats will be sent. The default value is 60 minutes.
- message_log - The path and name of the file to log messages related to LMU.

- Run the customization utility:

LMUCUSTW NETWARE

If you are in an OS/2 LAN Server environment, change the parameter NETWARE to IBM.

This utility will:

- Update the PATH environment variable located in the autoexec.bat file.
- Update the windowwin.ini file to automatically execute the LMUCLIWR command (LMUCLIWB, if installing for OS/2 LAN Server), by placing this command in the RUN statement.
- Create the windowslmu.ini file (if it does not exist), retrieving the information from a source file searched in the following order:
 - LMUW.INI
 - LMUW.CTL
 - LMU.CTL

If you are going to modify an existing configuration, first remove the windowslmu.ini file.

The station is now ready to be restarted.

8.2.4 LMU Configuration for NetWare Servers

Follow these steps to configure your NetWare server:

- In the /lmu2 directory, copy the lmubind.smp file to lmubind.ct1. In this new file, you are going to set the values for:
 - managing_system - The internetwork address of the OS/2 machine that will monitor the heartbeat of this server.
 - managing_system_with_database - The internetwork address of the OS/2 machine that will maintain the LMU database. It can be the same station as the MANAGING_SYSTEM.
 - fault_manager - The internetwork address of the OS/2 machine acting as the alert handler.
 - message_log - The volume, path and file name of the LMU message log on this server.
 - pulse_rate - The interval in minutes between the heartbeats. This value is specified using 4 hexadecimal digits, with 0000 meaning only initialization and termination heartbeats.
- At the NetWare server console, run the following commands:
 - load <volume:path>LMUBNDCS.NLM

This module will update the bindery (NetWare internal database) with the LMU control variables provided by the lmubind.ct1 file.

 - load <volume:path>LMUNLMCS.NLM

This module updates the bindery with the location of the `autoexec.bat` file and inserts the following line in this file:

```
c:\qdosvpd.com +kc: +Q
```

This command stores the user-provided data in the C drive in quiet mode. This data will be used by the `QUERYVPD.NLM`.

- `load <volume:path>LMUNCFCS.NLM`

The `autoexec.ncf` file (one of the files used in the server startup) is appended with these lines:

```
SEARCH ADD SYS:LMU2  
LOAD QUERYVPD /R  
LOAD LMUCLI
```

This will add the `LMU2` subdirectory to the search path, send the user-provided data to the managing system maintaining the database and load the `LMU` client code to receive remote commands. It will also send a heartbeat to the managing system.

- `load <volume:path>LMUVPDCS.NLM`

Copies the `qdosvpd.com` and the `adapters.tbl` files to the DOS partition in the same path as the `autoexec.bat` file.

- Restart the machine.

After this customization, a utility called `LMULOAD` will be available at the servers console that enables you to view and modify the parameters defined in the `lmubind.ctl` file in interactive mode. Be careful while using this utility because it considers a space as a valid entry. If you accidentally press the space bar, it will overwrite the previous information and store a space in its place.

8.3 LMU Startup

Since there is a specific sequence for the software components to be started in order to have the `LMU` environment available, it is advisable to check the setup of your system.

There are some phases that need intervention and you might want to disable the automatic startup procedure. These procedures are located in the `startup.cmd` file and in the startup folder under the `OS/2` System icon.

There are also some procedures that are automatically started by the `OS/2` system to place the machine in the same state it was in when it was last shut down.

The sequence we found to be effective in our environment (administrator, managing, managed and proxy agent station maintaining the `LMU` database) was:

- Start `TCP/IP`

From an `OS/2` command prompt type:

```
TCPSTART
```

or use the `OS/2` Startup folder.

- Start the `NetView` for `OS/2` agents:

After installing NetView for OS/2, two icons were placed in your Startup folder. You may want to take these out and start them manually. It is recommended to always start the NetView for OS/2 agents before starting the NetView for OS/2 daemons.

Start the agents by double-clicking on the NetView for OS/2 Agents icon.

- Log on to your LAN

If you are running a NetWare LAN, log in as you normally do. For example:
login supervisor

If you are running an OS/2 LAN Server LAN, log on as you normally do. For example:

```
logon LMUMGR /P:xxxxxx /D:nvsrvdm /V:d
```

- Ensure that DB2/2 has been started.

For example:

```
STARTDBM
```

- Log on to the local database environment.

For example:

```
logon USERID /p:PASSWORD
```

This logon is necessary to have access to the DB2/2 database. You can enter this command at any OS/2 prompt, or from the User Profile Management Services folder. Double click on the LOGON icon and log on as the admin user who has created the LMU database.

- Start LMU:

From an OS/2 prompt, type:

```
LMUSTART
```

Note: You can also put these commands in the above order in your STARTUP.CMD file. The startup file that we used is shown in 2.7.4, “Starting a NetView for OS/2 Managing System with the LMU Proxy Agent” on page 43.

If you are starting the LMU server for the first time, it will build the LMU database. This operation takes a few minutes and you should receive messages similar to those shown in Figure 160. You will receive a message that LMUSRV is *Creating database LMU2*.

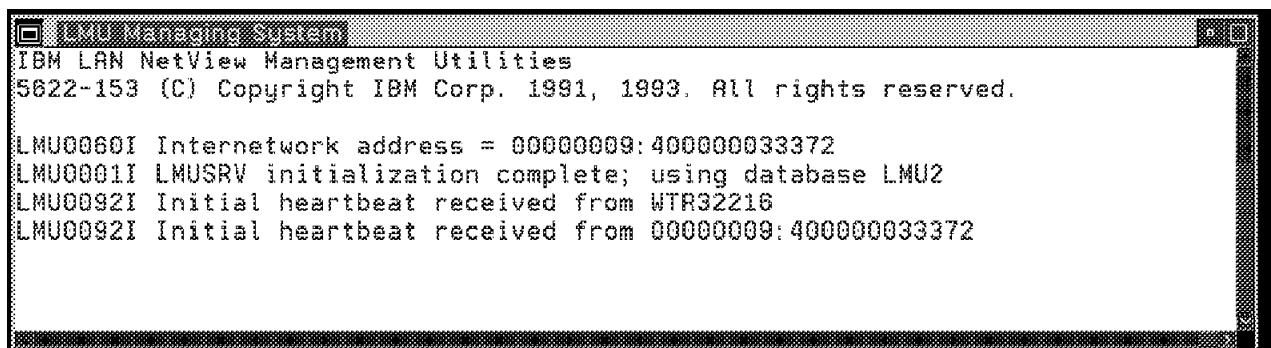


Figure 160. LMU Managing Station Panel Showing Initial Messages

While the database is being created, you may have problems starting the proxy agent as it will try to access a database that is still being built. In this case, the proxy agent fails. To restart it type the following command:

```
start "LMU SNMP proxy agent" /C d:\mu2\musnmpd.exe /d nv2mgr1 public
```

Note: Do not forget to change the parameters *nv2mgr1* and *public* to match your SNMP agent station name and community name, respectively.

If the proxy agent starts correctly you will receive the messages shown in Figure 161, in the LMU SNMP Proxy Agent window and the SNMPD window.

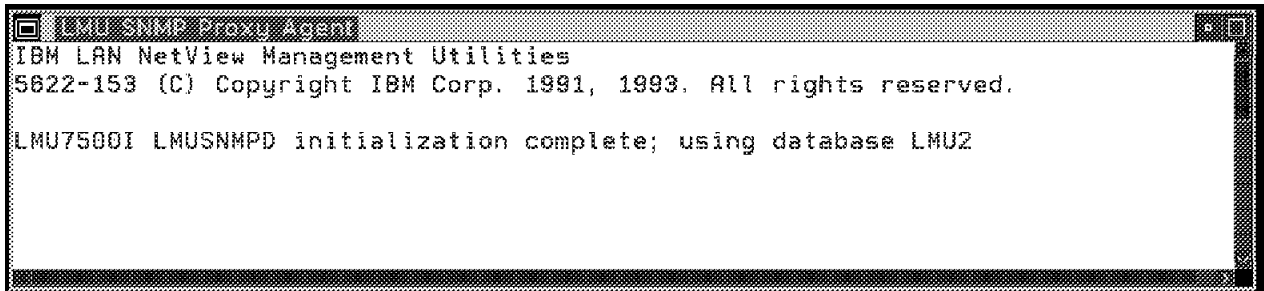


Figure 161. LMU Proxy Agent and SNMPD Panel Showing Initial Messages

When the environment is up and running, you can populate the LMU database with all the hardware and software configuration data and with the information you have provided in the *uservpd.dat* file for the OS/2 managed system. This is done by issuing the command:

```
QUERYVPD /R
```

You can request that the other managed stations that you have configured (DOS, Windows, NetWare server) send their *uservpd.dat* data as well. Refer to each configuration session to learn how to do that manually.

In addition, you can use the scheduler function of LMU to have the vital product data sent to the managing systems database on a regular basis. In our environment, we scheduled the command to be issued to all system's at midnight each day.

8.4 LMU Remote Commands from NetView for OS/2

Remote commands can be issued to LMU managed nodes through the NetView for OS/2 Management Desk interface. The TCP/IP REXEC command is used to send remote commands to the LMU proxy agent, for execution on the managed systems. To automate this process, a NETRC file can be built that gives the internet address, the REXEC username and password of the LMU proxy agent workstation. This file should be placed in the \TCPIP\ETC directory on the NetView for OS/2 managing workstation.

The contents of the NETRC file should like this:

```
machine 9.24.104.54 login fred password xxxxxx
```

Important

If NetView for OS/2 will be issuing remote commands for the LMU managing and managed systems through the LMU SNMP proxy agent, the proxy agent workstation must be an LMU administrator workstation.

8.5 Integrating LMU Alerts into NetView for OS/2

In this section, we will use a scenario to show how you can use some of the LMU utilities to manage your remote systems, and then integrate these utilities into a management platform such as NetView for OS/2.

8.5.1 Outlining the Systems Management Scenario

In this scenario, we will show the reader how to:

- Integrate both the LMU agent and the NetView for OS/2 agents.
- Use the LMU APPWATCH utility to monitor applications.
- Have LMU utilities send alerts to NetView for OS/2.
- Have the NetView for OS/2 Event Automator handle alert automation and resolution.
- Use the LMUCMD to send commands to remote stations.

What we are going to show is the APPWATCH utility running on a remote server, watching to ensure that both the OS/2 LAN Server Program (NETSERVER.EXE) is running, and that all the NetView for OS/2 agents are running. If any of the processes stop when they are not supposed to, APPWATCH will send an alert to the LMU proxy agent.

The proxy agent will translate the alert to an SNMP trap and forward it to a NetView for OS/2 managing station. In our case, both were on the same machine but the process is the same as if they were on two separate machines.

Once the trap is received by the NetView for OS/2 managing station, it is shown on the Event Displayer and accessed by the Event Automator. The Event Automator will display a system-supplied pop-up message to the console operator and then invoke a REXX command file that will determine which process failed on the remote machine. This REXX program will then send a command to the remote machine to attempt to restart the failed process.

8.5.2 Setting Up the LMU Managed Station

This section will show the reader how to first install and customize an LMU OS/2 managed machine and then show what is required to set up the APPWATCH utility.

8.5.2.1 Installing and Customizing the Managed Station

You can install and customize an OS/2 LMU managed station in three easy steps:

1. Install the LMU code on the OS/2 workstation.

Insert the LMU diskette 1 in your A-drive and type the following command:

A:\LMUINST IBM /Td

where *IBM* means to install the OS/2 LMU client code and *d* is the target drive to install on.

2. Update the LMU.CTL Information.

There are only 5 parts in the LMU.CTL that you have to update. To make things easy, you can take a copy of the LMU.CTL file that you created for the LMU managing station in 8.2.1.2, "LMU Control Files Configuration" on page 160, and use it for the managed station. It should still point to the proper managing station which will also have the SNMP proxy agent running on it.

The 5 sections you have to update are:

- a. MANAGING_SYSTEM
- b. MANAGING_SYSTEM_WITH_DATABASE
- c. FAULT_MANAGER
- d. GRAPHICAL_USER_INTERFACE
- e. APPWATCH_TABLE

The rest of the control keys will be ignored when running LMUCUST. For example, the section defining who your managing system is should look similar to this:

```
APP(LMU_UTILITY),  
    KEY(MANAGING_SYSTEM),  
    ASCIIIZ(WTR33372);
```

Note: We can use either the IPX address or the NetBIOS *computername* in this address field. The IPX address of our managing station was 00000009:400000033372 and the NetBIOS *computername* was WTR33372. In this case, we chose the NetBIOS name.

Also, make sure that the section defining your APPWATCH table looks like:

```
APP(LMU_UTILITY),  
    KEY(APPWATCH_TABLE),  
    ASCIIIZ(D:\LMU2\APPWATCH.TAB);
```

To see the entire LMU.CTL file that we used on the MANAGED station, please see A.3, "LMU/2 Control File (LMU.CTL) for the Managed System" on page 283.

3. Run the LMU customization program LMUCUST.

You need to run the LMUCUST program with the following parameters:

```
LMUCUST MANAGED FAULT_REPORTER /Td
```

where *d* is the target drive where LMU has been installed.

Running the LMUCUST program created an LMUSTART.CMD command file that you will use to start the LMU client on your OS/2 workstation. You will have something very similar to the file shown in Figure 162 on page 171.

```

@ECHO OFF
REM D:\LMU2\LMUCUST.EXE    Maintenance Level: LM00200
REM LMUCUST parameters: MANAGED FAULT_REPORTER /TD
START D:\LMU2\LMUCLI.EXE
DETACH D:\LMU2\APPWATCH.EXE /I0 /S2
EXIT

```

Figure 162. LMUSTART.CMD on the LMU Managed Machine

Note: LMUCUST did not put the detach command in the lmustart.cmd file. That line is needed for running the APPWATCH utility. Therefore, please add the following line to your LMUSTART.CMD file:

```
DETACH D:\LMU2\APPWATCH.EXE /I0 /S2
```

Later, you can start the LMU agent by typing lmustart.

Note: Do not start LMU at this time. We must first update the APPWATCH.TAB file as explained in 8.5.2.2, “Setting Up the APPWATCH Table.”

LMUSTART will also start the APPWATCH utility with two parameters:

1. **/I0** - an interval of time zero. This means that it will run forever. Since APPWATCH starts in DETACH mode, it will run as a background process.
2. **/S2** - it will sleep for 2 minutes and then check if all the processes identified in the APPWATCH.TAB table are up and running. It will check every 2 minutes.

8.5.2.2 Setting Up the APPWATCH Table

You will have to update the \LMU2\APPWATCH.TAB file. You will need to enter the names of the processes that you want to monitor. Monitoring in this case means that you want to make sure they are up and running. Since we want to be monitoring the OS/2 LAN Server program and the four NetView for OS/2 agent daemons, we would put a line item into the \LMU2\APPWATCH.TAB file for each of the 5 following processes:

- NETSERVER.EXE - LAN Server program
- LRAGENT.EXE - NetView for OS/2 LAN Requester agent
- LSAGENT.EXE - NetView for OS/2 LAN Server agent
- OS2_SIA.EXE - NetView for OS/2 System information agent
- SNMPD.EXE - NetView for OS/2 SNMP multiprotocol daemon

Following is the \LMU2\APPWATCH.TAB file after customization as shown in Figure 163 on page 172.

```

#
# Sample application table
#
#  Appl Name      Minimum      Maximum      Threshold      Reset/Accumulate
#  -----
#  LRAGENT.EXE    1          1          1          R
#  LSAGENT.EXE    1          1          1          R
#  SIA_OS2.EXE    1          1          1          R
#  SNMPD.EXE      1          1          1          R
#  NETSERVER.EXE  2          2          1          R
#
# Where:
#  Appl Name      Application name:
#                  - For OS/2 1.x this is the internal process name
#                    of the application (1 to 8 characters);
#                  - For OS/2 2.x this is the file name (including
#                    extension) of the application (1 to 12 characters)
#  Minimum        Minimum instances of application; range is 0 to 256;
#                  default is 1
#  Maximum        Maximum instances of application; range is 0 to 256;
#                  default is 1
#  Threshold      Number of violations (process instances not between
#                  minimum and maximum) before an alert is sent;
#                  range is 1 128; default is 1
#  Reset/Accumulate RESET indicates that the violation count should
#                  be reset to zero after an alert is sent;
#                  ACCUMULATE indicates that the violation count
#                  should not be reset after an alert is sent;
#                  default is RESET
#
# Notes:
# - Only application name is required
# - Use an asterisk (*) as a place holder (default value of entry is used)
# - Table entries are order dependent but not column dependent
# - Separate table entries with one or more blanks
# - A non-EXE (device driver, dynamic link library, font, etc.) that
#   has been loaded constitutes one instance; otherwise the number of
#   instances is 0
#

```

Figure 163. APPWATCH.TAB Table on LMU Managed Machine

Once you have completed editing the above file, you can start the LMU client by issuing the following command at an OS/2 prompt:

```
LMUSTART
```

8.5.3 Setting Up the NetView for OS/2 Managing Station

Once the LMU agent code is running on the remote OS/2 LAN Server machine, you can set up the NetView for OS/2 managing system to receive traps from the SNMP proxy agent. You will have to set up the Event Automator, and then write the necessary REXX program to qualify the LMU alert. In addition, an LMU remote command will need to be issued to resolve the alert situation. The resolution could be as simple as restarting the failed process.

8.5.3.1 Setting Up Event Automator to Handle LMU Alerts

This section will show how to set up the Event Automator to handle alerts coming from an LMU proxy agent. For a more detailed explanation on setting up the Event Automator, please see 3.6, “Event Automation and the Event Displayer” on page 83.

If we double click on the **Event Automation** icon on the NetView for OS/2 Main Icon View window, we will get the *Event Automation Update* window as shown in Figure 164.

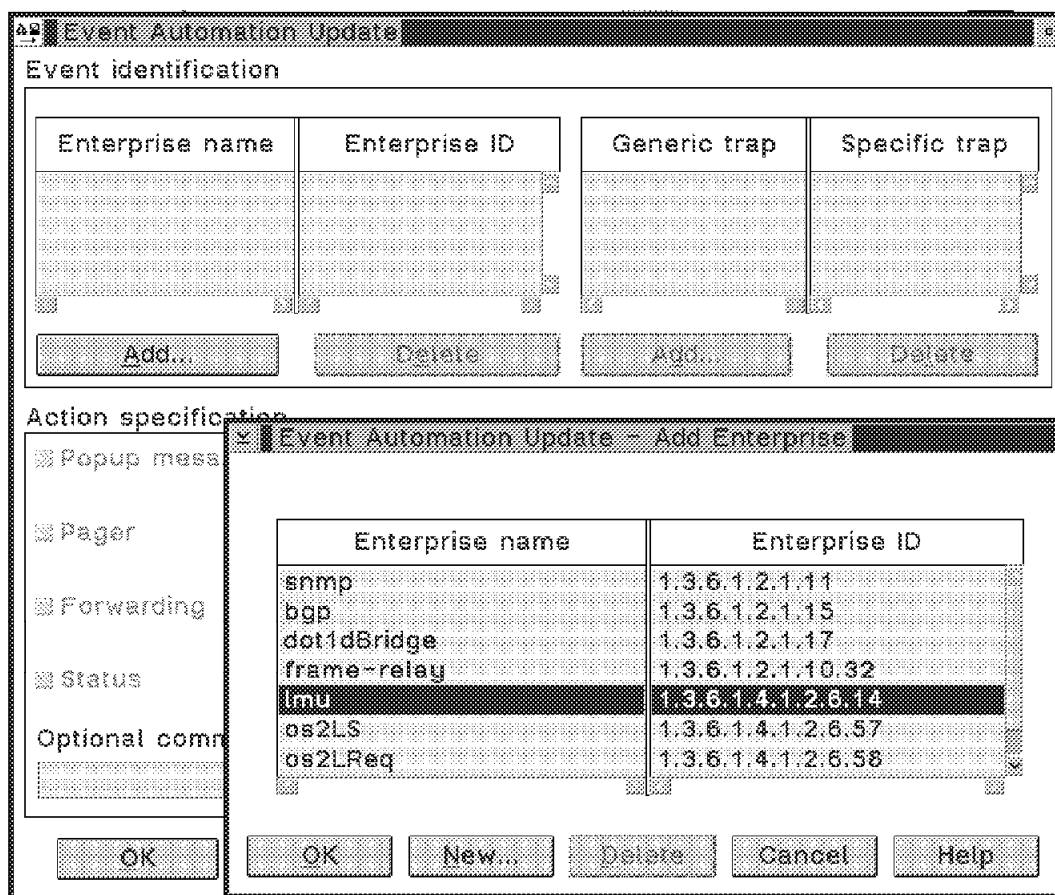


Figure 164. Event Automation Update Window - Adding LMU Type Alerts

If we click on the **Add...** button under the *Enterprise name* list box, we will get the *Event Automation Update - Add Enterprise* window as shown in Figure 164.

If we now click on the **lmv** line item and click on the **OK** button, it will add the *lmv* enterprise to the *Enterprise name* list box as shown in Figure 165 on page 174.

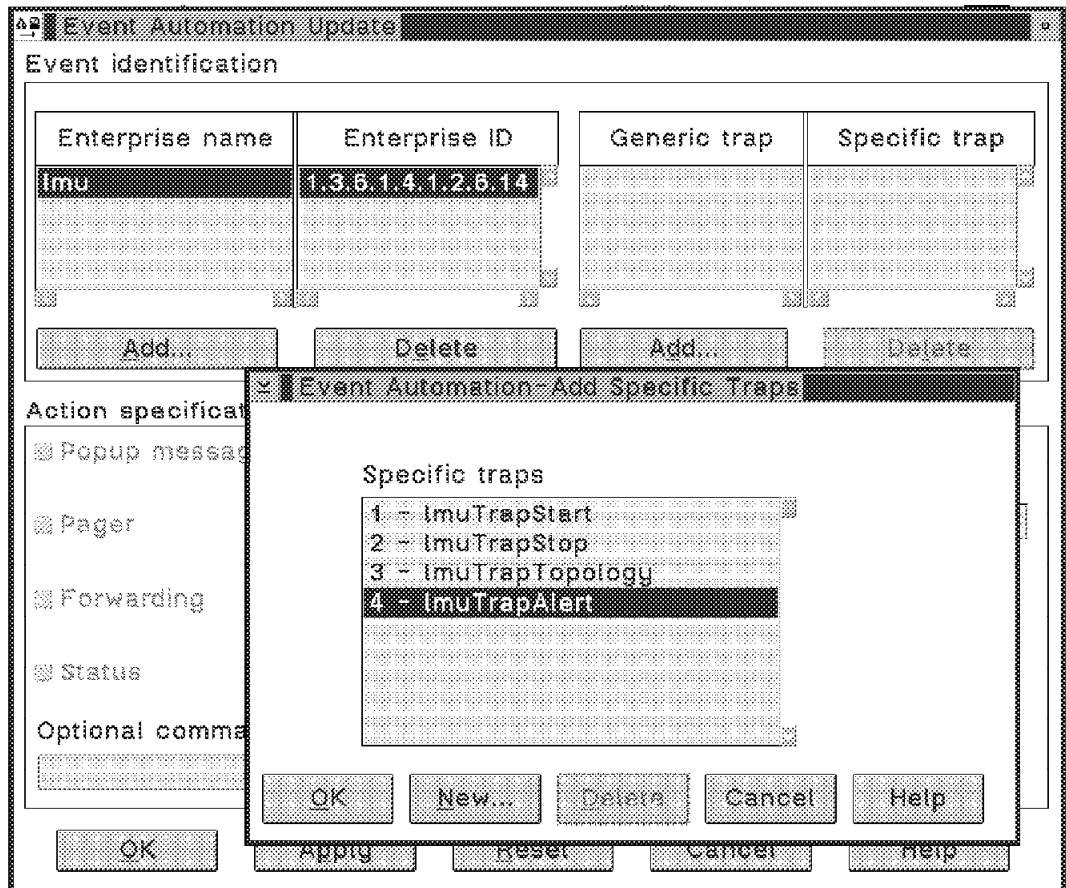


Figure 165. Adding Generic/Specific LMU Traps to the Event Automator

In Figure 165, we click on the **Imu** line item to highlight it and then click on the **Add...** button under the *Generic Trap* list box. We get the *Event Automation - Add Specific Trap* window as shown in Figure 165.

If we now click on the **ImuTrapAlert** line item and click on the **OK** button, it will add the *ImuTrapAlert* line item to the *Generic/Specific Trap* list box as shown in Figure 166 on page 175.

Event identification

Enterprise name	Enterprise ID	Generic trap	Specific trap
lmu	1.3.6.1.4.1.2.6.14	6 - enterpriseS	4 - lmuTrapAlc

Add... Delete Add... Delete

Action specification

☒ Popup message

☐ Pager Alias Message

☐ Forwarding Address

☐ Status New status

Optional command(s):

D:\ANV2\SNMPREXX\APPSTART.CMD Find...

OK Apply Reset Cancel Help

Figure 166. Specifying Actions Upon Receipt of the LMU Alert

Click on the trap line items that we just entered to ensure that they are highlighted as shown in Figure 166. This will make the check boxes and entry fields in the *Action Specification* area become active.

Click on the Popup message check box so that we get the system-supplied pop-up for the LMU trap.

We will put the name of our REXX program in the Optional command(s) field. The command that we entered was:

D:\ANV2\SNMPREXX\APPSTART.CMD

We next clicked on the **OK** button to save our Event Automator entry. It now becomes active.

8.5.3.2 Writing the REXX Program to Restart Remote Processes

At this point, it is necessary to write the REXX program that will run as a result of the LMU trap coming in and notify our managing station of some failed process on some machine.

When an LMU alert arrives at the LMU managing station, the SNMP proxy agent residing there translates the alert into an SNMP trap and sends it to the NetView for OS/2 managing station. When it arrives at the NetView for OS/2 managing station, it is stored in a special lmu MIB. We can then access this MIB using the MIB Browser to view the alerts that have come from LMU.

By using the MIB Browser, we can see in what format the alert arrived and what informational message was in the trap. Using REXX, we can duplicate the SNMP GET command and parse the output into a format that is useful in our scenario.

The MIB Variable that stores the LMU alert text is *ImuAlertUserText*. To get to this MIB variable, parse the following MIB tree:

```
* private
  enterprises
    ibm
      ibmProd
        Imu
          ImuAlert
            ImuAlertTable
              ImuAlertTableEntry
                ImuAlertUserText
```

Using the MIB Browser, we ran a query on the *ImuAlertUserText* variable and got the result shown in Figure 167:

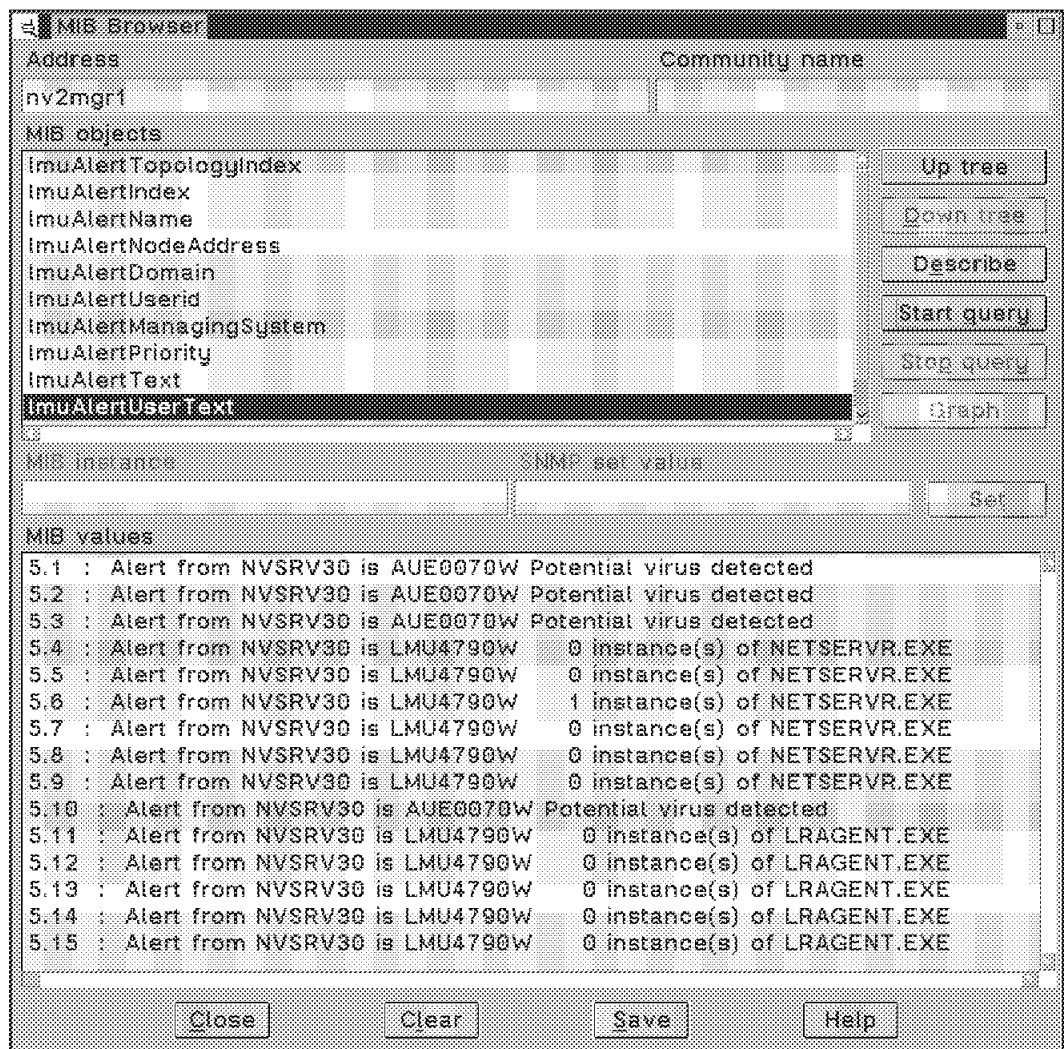


Figure 167. MIB Browser Showing the Format of LMU Alerts

Notice that the alert message tells us who the alert came from, what the LMU message number is for APPWATCH alerts, and what process has failed (for example, 0 instances found).

We can now write a REXX routine to do an SNMPWALK of this *ImuAlertUserText* variable and if it is one of the processes that we are monitoring in the APPWATCH table, then we can potentially restart that process. The REXX program that does this is shown in Figure 168 on page 178.

```

/*REXX*/
'@echo off'
/* This program will detect what kind of LMU alert has come into */
/* the NetView for OS/2 Managing Station and then perform the */
/* appropriate action. It checks to see if an APPWATCH alert has */
/* come in from a server. It then checks to see what application */
/* has gone down, and then restarts that application. */
/* Author: Dan Heimann IBM Canada */
/* */
/*****
/* Call the SNMPWALK command to retrieve information from the */
/* lmuAlertUserText MIB Variable. This is essentially a log of all */
/* received LMU Alerts */
/*****
FQ_Obj_Id1=".iso.org.dod.internet.private.enterprises.ibm.ibmProd."
FQ_Obj_Id2="lmu.lmuAlert.lmuAlertTable.lmuAlertEntry.lmuAlertUserText"
lmu_Object_Id = FQ_Obj_Id1 || FQ_Obj_Id2
'call snmpcmd snmpwalk nv2mgr1 'lmu_Object_Id' rxqueue /lifo'
/*****
/* Pull off the most recent LMU Alert to see if it is an */
/* APPWATCH Alert saying that "0 instance(s) of xxxxxxxx.EXE" */
/*****
pull line
parse var line mibinstance ":" instancetype ":" AlertText
parse var AlertText . . ServerName . LmuMsgNo LmuMsgText
/*****
/* If the LMU Message Number is for APPWATCH alerts, then check */
/* for zero instances of a particular application */
/*****
IF LmuMsgNo = 'LMU4790W' THEN
DO
  parse var LmuMsgText numb . . AppName
  IF numb = 0 THEN
    SELECT
      /* */
      /* Start the NetView Agents if they are not running */
      /* */
      WHEN AppName=' LRAGENT.EXE' THEN
        'LMUCMD /Q 'ServerName' START "LAN Requester Agent" /MIN LRAGENT.EXE'
      WHEN AppName=' LSAGENT.EXE' THEN
        'LMUCMD /Q 'ServerName' START "LAN Server Agent" /MIN LSAGENT.EXE'
      WHEN AppName=' OS2_SIA.EXE' THEN
        'LMUCMD /Q 'ServerName' START "System Information Agent" /MIN OS2_SIA.EXE'
      WHEN AppName=' SNMPMPD.EXE' THEN
        'LMUCMD /Q 'ServerName' START "SNMP Multi-Protocol Daemon" /MIN SNMPMPD.EXE'
      /* */
      /* Start OS/2 LAN Server if it is not running */
      /* */
      WHEN AppName=' NETSERVR.EXE' THEN 'LMUCMD /Q 'ServerName' NET START SERVER'
      /* */
      /* Add your own Applications here */
      OTHERWISE Nop
    END
  END
EXIT

```

Figure 168. APPSTART REXX Command File

8.5.4 Generating and Resolving the Alert Situation

We are now ready to test our scenario. If we are at the LMU managing station with the proxy agent running on it we can start the LMUGUI by clicking on the LMU GUI icon on the LMU Main Icon View window, or by typing `start lmugui` from an OS/2 window. LMU will dynamically build a network topology of all discovered LMU agents and present them on the *LMU GUI - Managing System and Node View* window as shown in Figure 169.

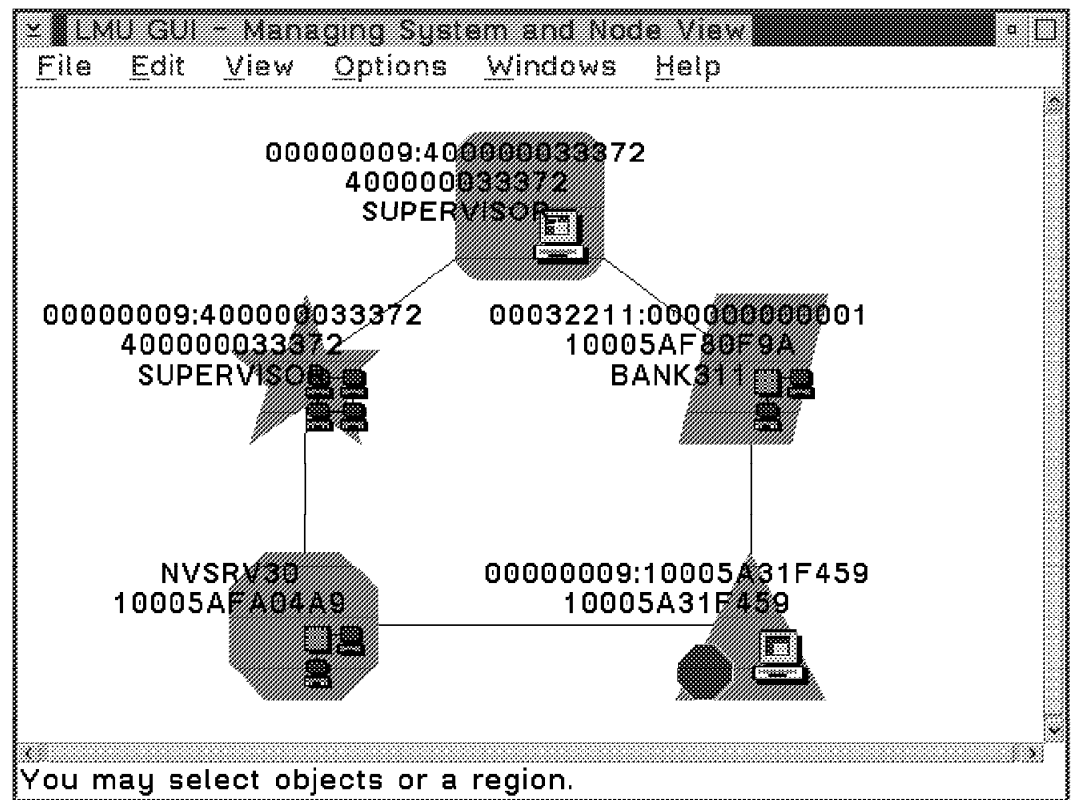


Figure 169. LMU GUI - Managing System and Node View

Figure 169 shows all of the nodes in their *active* state. In our scenario, the NetView for OS/2 LAN Requester agent fails on the machine having the NetBIOS computername: *NVSRV30*. Since it is an LMU client and it's running the APPWATCH utility, it will know within 2 minutes that the LAN Requester agent has gone down. Once this happens, it sends an alert to its LMU managing station which is identified on the GUI as *00000009:400000033372*. When the alert arrives, the GUI gives several audible beeps and puts hash marks across the affected icons as shown in Figure 170 on page 180.

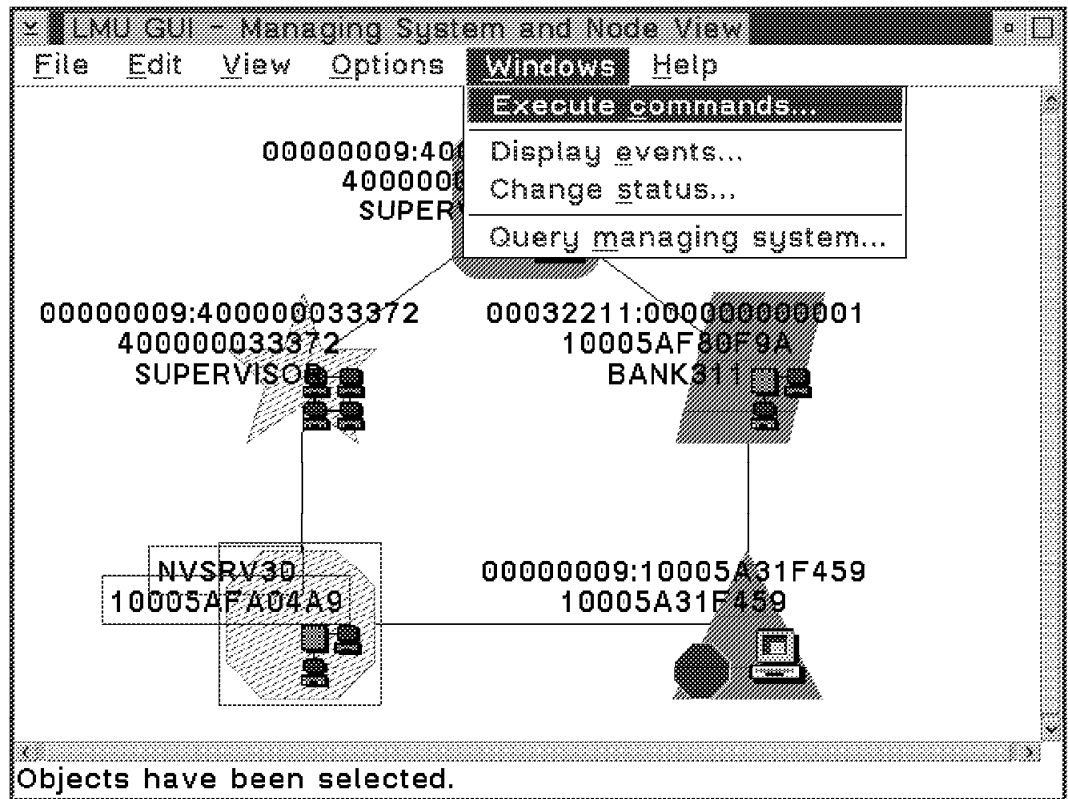


Figure 170. LMU GUI - APPWATCH Alert Has Arrived - Display Events

If we want to view the alert text from the LMU GUI, we click on **Windows** on the action bar, and then on the **Display events...** line item in the drop box. This will show us the *LMU GUI - Events for NVSRV30* window as shown in Figure 171.

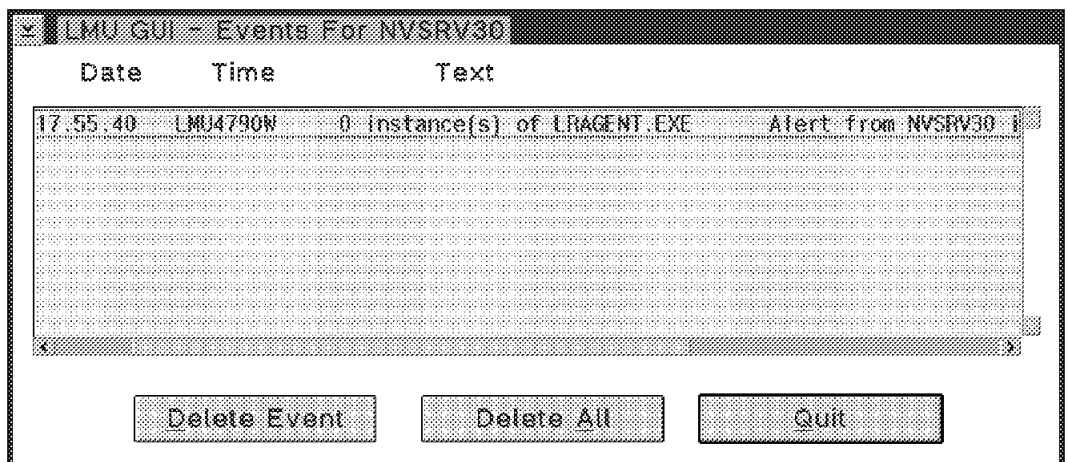


Figure 171. Viewing the Alert from the LMU GUI Display Events Window

At this point, the SNMP proxy agent will take a copy of the alert, translate it into an SNMP trap and forward it to the NetView for OS/2 managing station. The trap destination must have been set up as shown in Figure 157 on page 160 and described in 8.2.1.1, "IBM TCP/IP for OS/2 Configuration" on page 158.

Once the trap has arrived at the NetView for OS/2 managing station, it can be viewed using the MIB Browser or you can use the Event Displayer. To use the

Event Display, double click on the **Event Displayer** icon in the NetView for OS/2 Main Icon View window and you will see the window shown in Figure 172 on page 181.

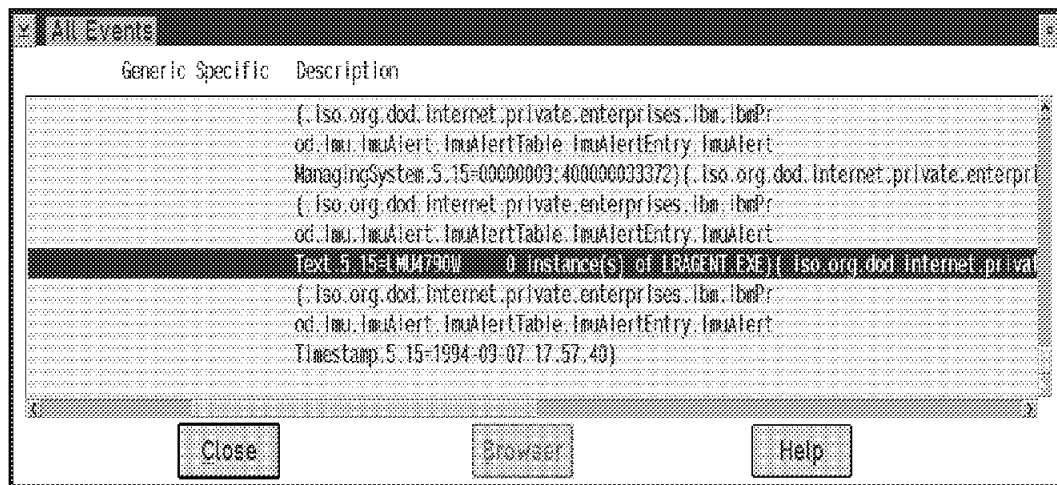


Figure 172. Using NetView for OS/2 Event Displayer to Show LMU Alerts

Note: We can see that the LMU alert message highlighted in the previous window is basically the same message that you saw in the LMU Display Events window.

Now the last part of the scenario can take place. Besides displaying the alert in the Event Displayer window, the alert is also sent to the Event Automator. Remember that we have chosen to display a system-supplied pop-up message, and to invoke our APPSTART REXX routine to restart the LAN Requester agent that went down. As soon as the alert came to the NetView for OS/2 managing system, the system-supplied pop-up was displayed. We have captured this pop-up message as an inset to the LMU GUI and it is shown in Figure 173 on page 182.

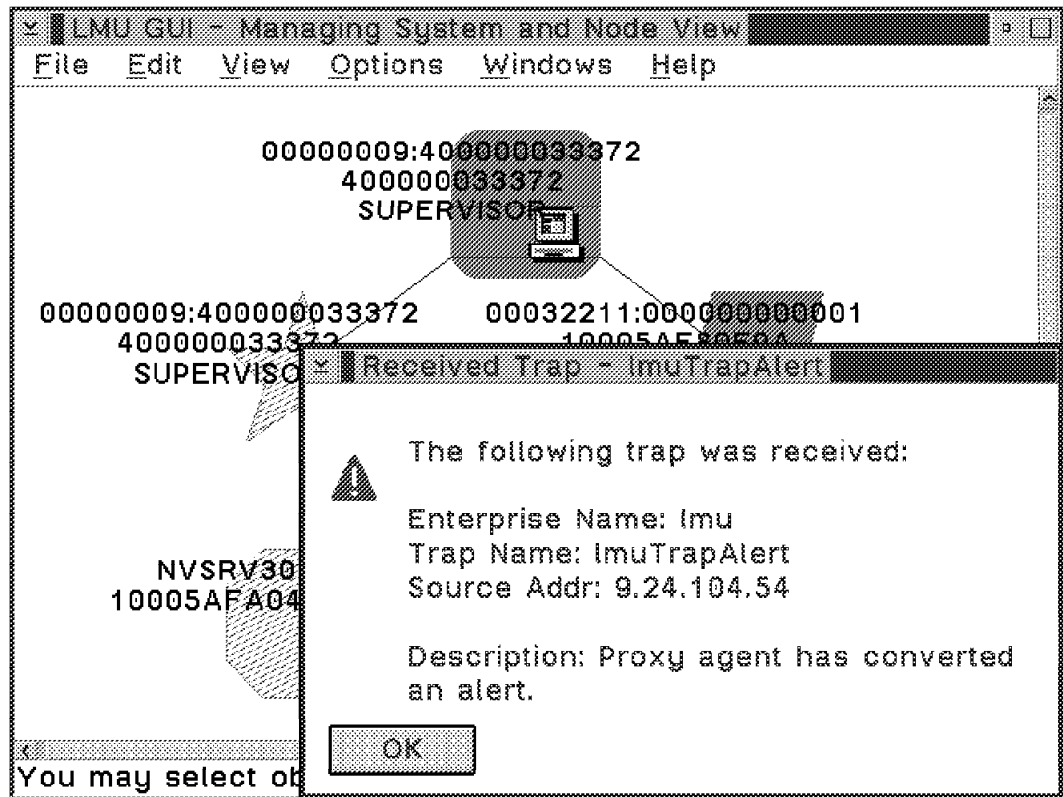


Figure 173. Event Automator - Default Pop-Up for LMU Traps Received

We can show that the pop-up was displayed but there is nothing that we can show regarding the APPSTART REXX program. It did in fact, parse the LMU alert text that it got from `ImuAlertUserText`. It found that it was an APPWATCH alert, and routed a command using `LMUCMD` to restart the LAN Requester agent on the NVSRV30 machine. All of these actions were completely automated with no human operator intervention.

You can use the APPSTART routine in your shop to ensure that your mission critical applications stay up.

8.6 Gathering Vital Product Data (VPD)

At the time of writing this redbook, the SCAN function was not available. To get a machine's Vital Product Data, you can use the facilities of LMU to gather this VPD and store it in a central DB2/2 database. Following is a sample report as a result of running the following command:

```
QUERYVPD
```

To have the data stored in a DB2/2 database on the LMU managing station, execute the following command:

```
QUERYVPD /R
```

The `/R` parameter specifies to store the data in the relational database that is being used by LMU.

The following is an example of the output from the `QUERYVPD` command:

IBM LAN NetView Management Utilities
5622-153 (C) Copyright IBM Corp. 1991, 1993. All rights reserved.

IBM LAN NetView Management Utilities Maintenance Level LM00200

Report Generation

Date of report 1994-08-19
Time of report 15.56.27

Operating Environment

Operating System OS/2 2.11

Hardware Configuration

Machine Type IBM PS/2 Model 80
Product Number 8580-071
 Model ID F8
 Sub-Model ID 00
 Revision 00
 BIOS Date 03/30/1987
 Planar ID FEFF - PS/2 Model 80 16Mhz Planar
 ROM Copyright COPR IBM 1981 1987
Processor Intel 80386
 Processor Speed 16 MHz
 Processor Code 03
 Stepping Code D0
CoProcessor Intel 80387
Bus Type Micro Channel 32-Bit

Total Memory 14336 KB = 14.0 MB
 Conventional 640 KB
 Extended Memory 13696 KB

Equipment List 1 Parallel Port(s)
 1 Serial Port(s)
 1 Diskette Drive(s)
 2 Fixed Disk(s)
 Math CoProcessor

Serial Port 1 COM1
 Baud rate 1200 bps
 Data bits 7
 Parity Even
 Stop bits 1

Diskette Drive 1 3.50" - 1474K - 80 Tracks - Type 4

Fixed Disk 1 70 MB = 71680 KB = 73400320 bytes
 Cylinders 70
 Sectors 32
 Heads 64

```

Total Sectors          143360

Fixed Disk 2 ..... 70 MB = 71680 KB = 73400320 bytes
  Cylinders          70
  Sectors            32
  Heads              64
  Total Sectors      143360

Keyboard Type ..... 101/102 Key Enhanced (ID AB41)

Current Display ..... PS/2 Color 8514
  Current Video      BGA
  Video Memory       1024K

Primary Node Address ... 10005A22396F
Primary Universal ..... 10005A22396F

Total Slots ..... 8
  System            1
  User Slots        7

Expansion Slot 1 ..... IBM Enhanced Memory Adapter 80386
  Status            Enabled
  Identification    FDDF
  POS Data bytes    FF 02 07 FF
  Subaddress        00FF
  Starting Address  2 MB
  Installed Memory  12 MB
  SIMM Slot 1      4MB 80ns SIMM
  SIMM Slot 2      4MB 80ns SIMM
  SIMM Slot 3      4MB 80ns SIMM
  SIMM Slot 4      No memory

Expansion Slot 3 ..... IBM Token-Ring Network Adapter/A
  Status            Enabled
  Identification    E000
  POS Data bytes    C1 24 C8 C0
  Subaddress        0000
  Interrupt Level   2
  IO Address        A20-A23
  RAM Memory Address C0000-C3FFF
  ROM Memory Address C8000-C9FFF
  Adapter           Primary
  Adapter Speed     4 Mbps
  Burned-in Address 10005A22396F

Expansion Slot 6 ..... IBM Display Adapter 8514/A
  Status            Enabled
  Identification    EF7F
  POS Data bytes    01 FF FF FF
  Subaddress        FFFF

Expansion Slot 8 ..... IBM ESDI Fixed Disk Controller
  Status            Enabled
  Identification    DDFF
  POS Data bytes    55 17 FF FF
  Subaddress        FFFF

```

ROM Memory Address	DC000-DFFFF
IO Address	3518-351F
Arbitration Level	5
Interrupt Level	14

Logical Drives

HPFS	Drive C	6082 of	71664 KB free ==>	91% full
	Current directory	\		
	Volume	C_DRIVE		
HPFS	Drive D	12321 of	70640 KB free ==>	82% full
	Current directory	\		
	Volume	D_DRIVE		
FAT	Drive E	1423 of	1423 KB free ==>	0% full
	Current directory	\		
	Volume	VFDISK		
LAN	Drive U	937082 of	1185776 KB free ==>	20% full
	Network Name	\\WTRAS2\CIDX		
	Current directory	\		
	Volume	G_DRIVE		
LAN	Drive V	148556 of	162800 KB free ==>	8% full
	Network Name	\\WTRDC\LANCMDS		
	Current directory	\		
	Volume	IBM23MN465		
LAN	Drive W	148556 of	162800 KB free ==>	8% full
	Network Name	\\WTRDC\LANDLLS		
	Current directory	\		
	Volume	IBM23MN465		
LAN	Drive X	2228 of	195568 KB free ==>	98% full
	Network Name	\\WTRAS1\CID		
	Current directory	\		
	Volume	IBM9250556		
LAN	Drive Y	45461 of	194544 KB free ==>	76% full
	Network Name	\\WTRDC\JBARKER		
	Current directory	\		
	Volume	IBM23MN465		

User Data

Data Location D:\LMU2\USERVPD.DAT

(Barker,John)(123456)(333)(555-5555)(555-555-5555)(657)(001)(aa1
 2)(555-5555)(555-555-5555)()(XXX)(XXXXXXXXXX)(XXXXXX){IBM-8580
 ,071,23-1234567,09-01-1992}{IBM-8514,001,00-1097175,09-01-1992}{
 IBM-1391401,,4655517,09-01-1992}{IBM-90X6778,,1306518,09-01-1992
 }{IBM-4216,031,41-8886A,09-01-1992}{IBM-6180,,B1506,09-01-1992})

SYSLEVEL Information

IBM System Performance Monitor/2

Version	Version 2.00.1
Component ID	562201000
Current CSD Level	WR06075
Prior CSD Level	WR06075

IBM OS/2 First Failure Support Technology/2

Version	Version 1.20
Component ID	562119400
Current CSD Level	WR00475
Prior CSD Level	WR00470

IBM OS/2 32-bit Graphics Engine

Version	Version 2.11
Component ID	562107701
Current CSD Level	XR06200
Prior CSD Level	XR06200

IBM OS/2 Base Operating System

Version	Version 2.11
Component ID	562107701
Current CSD Level	XR06200
Prior CSD Level	XR06200

IBM OS/2 User Profile Management

Version	Version 3.00
Component ID	562125302
Current CSD Level	WR07045
Prior CSD Level	WR07000

IBM OS/2 User Profile Management - Extended

Version	Version 3.00
Component ID	562125306
Current CSD Level	IP07045
Prior CSD Level	IP07000

IBM OS/2 LAN Server/Requester Product

Version	Version 3.00
Component ID	562125305
Current CSD Level	IP07000
Prior CSD Level	IP07000

IBM OS/2 LAN Requester

Version	Version 3.00
Component ID	562125301
Current CSD Level	IP07045
Prior CSD Level	IP07000

IBM OS/2 LAN Adapter and Protocol Support

Version	Version 2.20.2
Component ID	562125303
Current CSD Level	WR07045
Prior CSD Level	WR07000

IBM Communications Manager/2 IBM Internal Use Only

Version	Version 1.11
Component ID	2207800
Current CSD Level	WR06150
Prior CSD Level	WR00000

TCP/IP BASE for OS/2 2.0 and 2.1

Version	Version 2.00
Component ID	562208600
Current CSD Level	UN50382
Prior CSD Level	UN00000

IBM LAN NetView Management Utilities for OS/2

Version	Version 1.00
Component ID	562215301
Current CSD Level	LM00201
Prior CSD Level	LM00201

IBM NetView for OS/2

Version	Version 2.00
Component ID	562254600
Current CSD Level	UN0233

IBM NetView for OS/2 - Agent for OS/2

Version	Version 2.00
Component ID	562254600
Current CSD Level	UN0233

Critical File Information

List of files D:\LMU2\CRITFILE.DEF

C:\CONFIG.SYS

Location	C:\
Date	1994-08-26
Time	15.29.58
Size	3779 bytes

C:\STARTUP.CMD

Location	C:\
Date	1994-08-26
Time	15.30.46
Size	854 bytes

C:\CMLIB\WTRMODEL.CFG

Location	C:\CMLIB\
Date	1994-08-26
Time	17.58.48
Size	52320 bytes

Chapter 9. Connecting to NetView on MVS/ESA

In this chapter we will exchange information with:

- NetView on MVS/ESA
- NetView on OS/2

We will show how to set up the communications between the environments, illustrate trap flow, and provide an introduction into automation.

9.1 Connecting to NetView for MVS/ESA

In this section, we will show how to set up Communications Manager/2 on a NetView for OS/2 Managing System so that it will communicate with NetView for MVS. We will also show what configuration is required on the host in order to communicate with Communications Manager/2 in an APPC LU 6.2 environment.

After communications have been established, we will show how to:

- Start the NetView for OS/2 Host Connection program.
- Use and change alert filters.
- Use NetView for MVS to monitor SNMP devices.
- Automate actions from NetView for MVS using the RUNCMD facility.

9.1.1 Setting Up Communications Manager/2

This section will take the reader through a step-by-step approach to setting up Communications Manager/2 to communicate with NetView for MVS in an LU 6.2 (APPC) environment. A prerequisite for NetView for OS/2, is to have Communications Manager/2 Version 1.01 or higher. We were using Version 1.11.

Important Background Information

It is important to note that the parameters used in the following sequence of windows show what we entered for our network environment. Please refer to the *WTRMODEL.NDF* configuration file in Appendix B.1, "CM/2 Configuration File" on page 291 to match up the parameters entered on the windows to those parameters inserted by Communications Manager/2 into the NDF file.

In addition, our network configuration consisted of *two* host machines, each belonging to a different Network ID. The *local* node which we normally log on to is called *USIBMMK* and the node where NetView for MVS is installed is *USIBMRA*. Therefore, we wanted USIBMRA to be my partner-session in an APPC environment. If your local logon system is also the same system where host NetView resides, then these two network IDs would be the same and it should be much easier to configure.

To start off, click on the **Communications Manager/2** icon on your OS/2 Desktop and you will get the Communications Manager/2 Icon View as shown in Figure 174 on page 190.

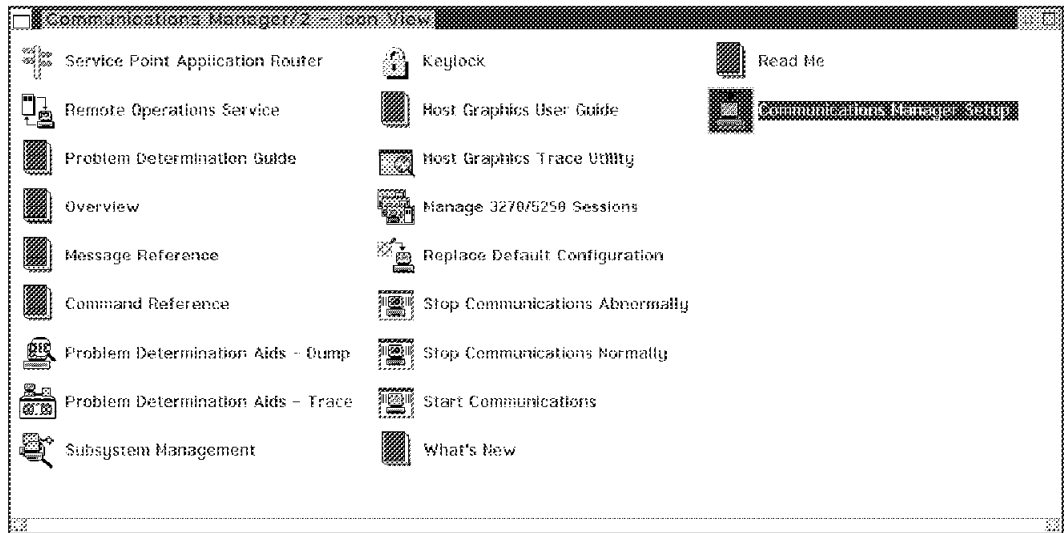


Figure 174. Communications Manager/2 - Icons/Functions Available

In Figure 174, we clicked on the **Communications Manager/2 Setup** icon and CM/2 presented us with the *Communications Manager/2 Setup* window as shown in Figure 175.

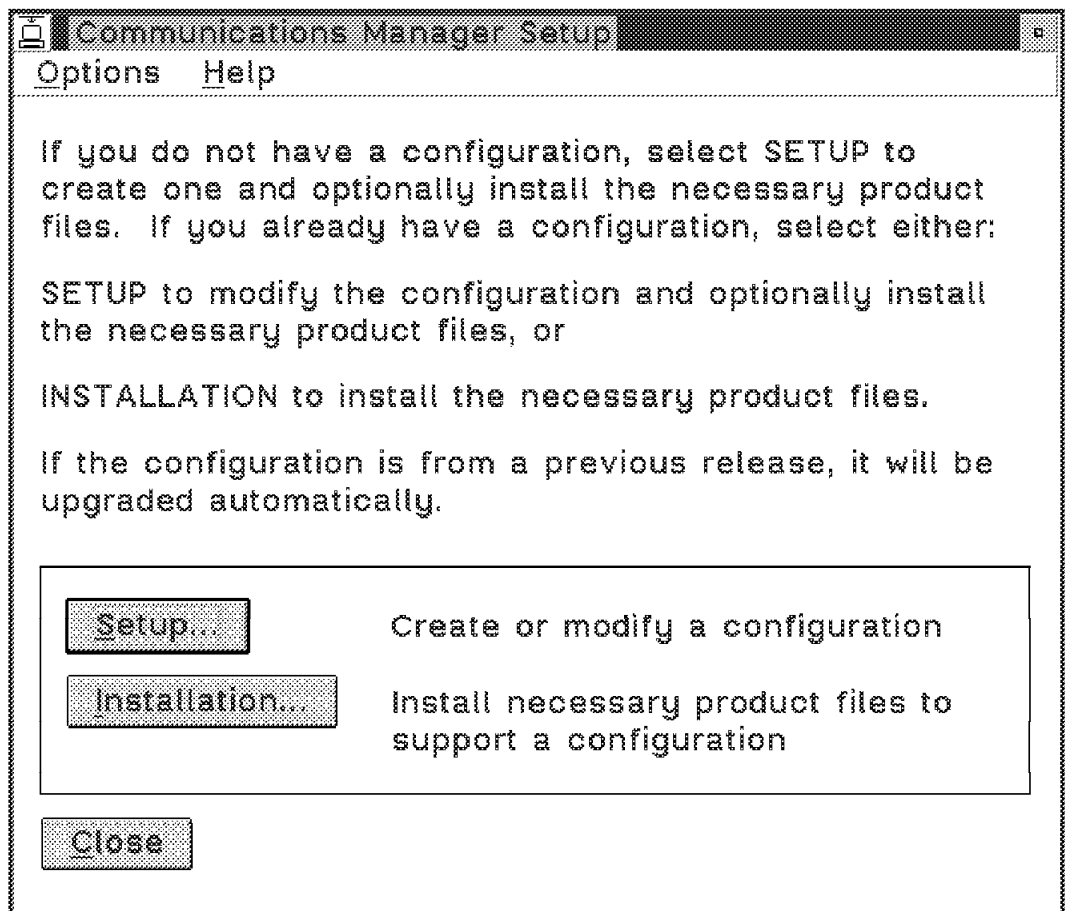


Figure 175. Communications Manager/2 Setup Window

In Figure 175, we clicked on the **Setup...** button and CM/2 presented us with the *Open Configuration* window as shown in Figure 176 on page 191.

The name of our configuration file is *WTRMODEL*. We had already set up our configuration and tested it so it will appear as the default.

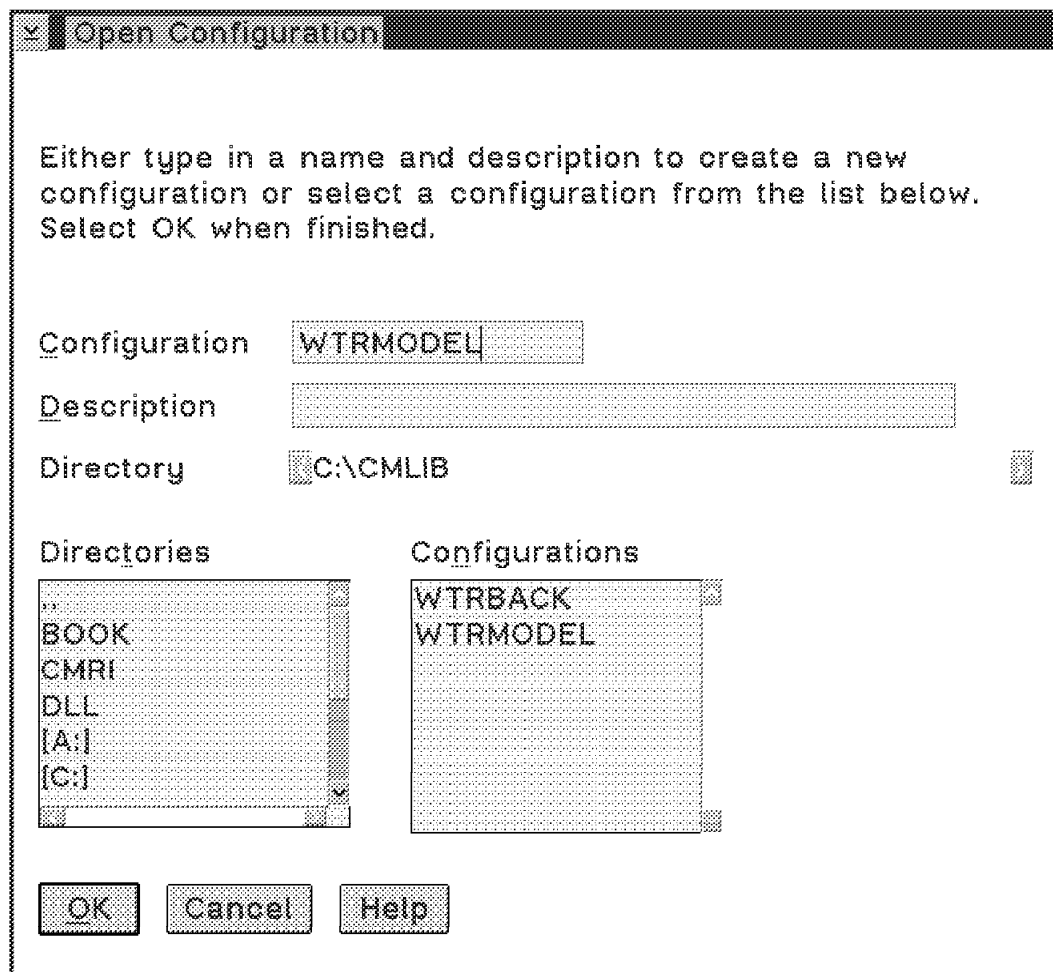


Figure 176. Communications Manager/2 - Open Configuration Window

If your *Configuration* field is empty, type in a new configuration file name and *Description* and click on the **OK** button. You will then be presented with the CM/2 Configuration Definition window as shown in Figure 177 on page 192.

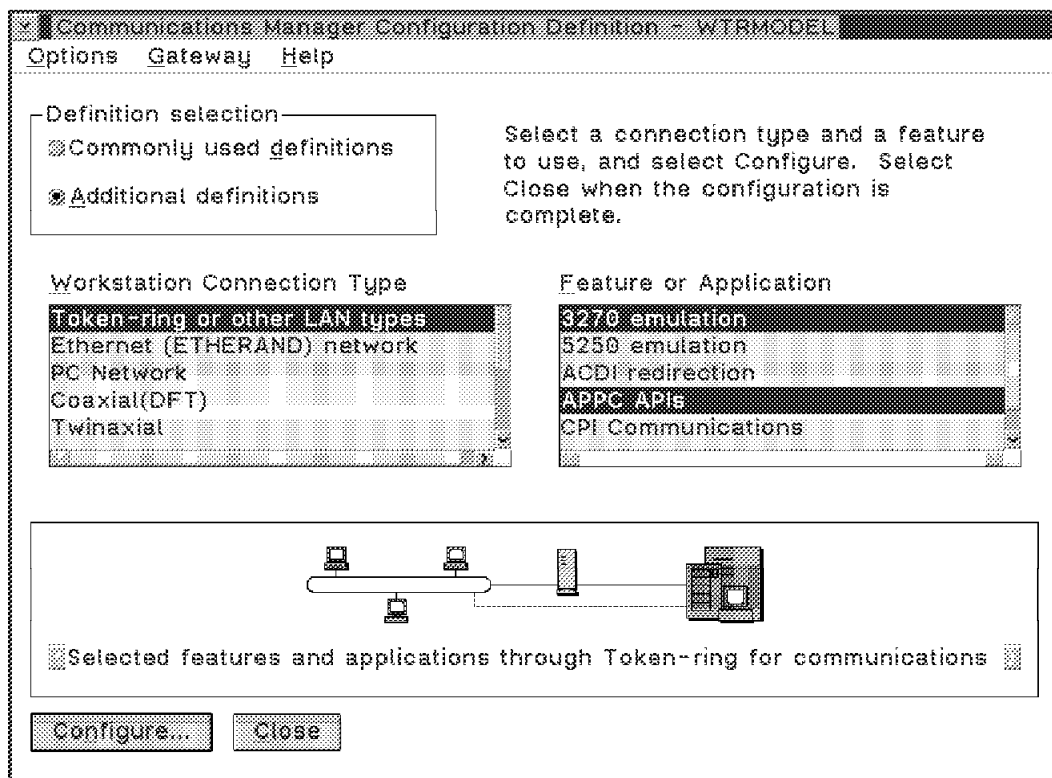


Figure 177. Communications Manager/2 - Configuration Definition for WTRMODEL

To keep things simple we selected the **Additional definitions** radio button and we were presented with the two scroll boxes shown in Figure 177. One was for **Workstation Connection Type**, and the other one was for **Feature or Application**.

Since we were using a token-ring, we clicked on the **Token-ring or other LAN types** as a workstation connection type, and on **3270 emulation** as a feature or application.

Note: If you already have 3270 emulation configured on your workstation, then you do *not* have to select it for configuration now.

To select more than one Feature or Application, such as the APPC APIs (which is mandatory for LU 6.2 communication to the host), press and hold the Ctrl key while clicking on the **APPC APIs** line item.

Next click on the **Configure...** button to configure each profile. You will be shown the Communications Manager/2 Profile List window as shown in Figure 178 on page 193.

9.1.1.1 Token-Ring LAN - DLC Adapter Parameters

The first thing that we will configure is the Data Link Control (DLC) parameters for token-ring. We click on the **DLC - Token-ring or other LAN types** line item as shown in Figure 178 on page 193.

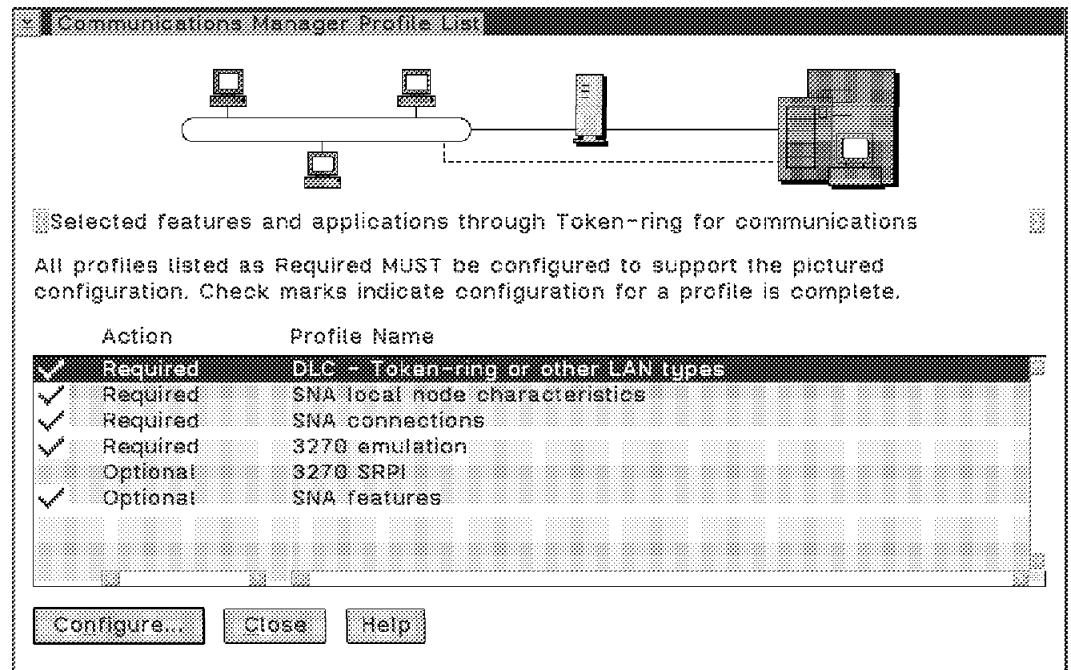


Figure 178. Communications Manager/2 Profile Listing

Press the **Configure...** button on the previous window and you will be presented with the actual DLC adapter parameter entry window as shown in Figure 179 on page 194.

Token Ring or Other LAN Types DLC Adapter Parameters

Adapter

☒ Free unused links

☒ Send alert for beaconing

☒ Maximum activation attempts (1 - 99)

Maximum link stations (1 - 255)

Maximum J-field size (265 - 16393)

Percent of incoming calls (%) (0 - 100)

Link establishment retransmission count (1 - 127)

Retransmission threshold (1 - 127)

Local sap (hex) (04 - 9C)

C&SM LAN ID

Connection network name (optional) .

Window count

Send window count (1 - 8)

Receive window count (1 - 8)

Figure 179. Setting LAN DLC Adapter Parameters

We used all of the defaults for this window except the *Maximum link stations*. The default is 4 but we entered 16 because NetView for OS/2 requires more link stations for NetBIOS protocol support. You need one link station for every logical link that your system uses. We then entered USIBMRA as our C&SM LAN ID. This is the Communications and Systems Management LAN ID that identifies the LAN that our adapter is on. You will need to get this network ID parameter from your on-site LAN network administrator.

Note: In our case, it just happened to have the same name as our partner network ID for our host NetView. The two are independent of each other.

Once you have entered this parameter, click on the **OK** button which will take you back to the CM/2 Profile List window as shown in Figure 180 on page 195.

9.1.1.2 SNA Local Node Definitions

We can now move to the next definition to be configured which is the SNA local node characteristic. Click on the **SNA local node characteristics** line item as shown in Figure 180.

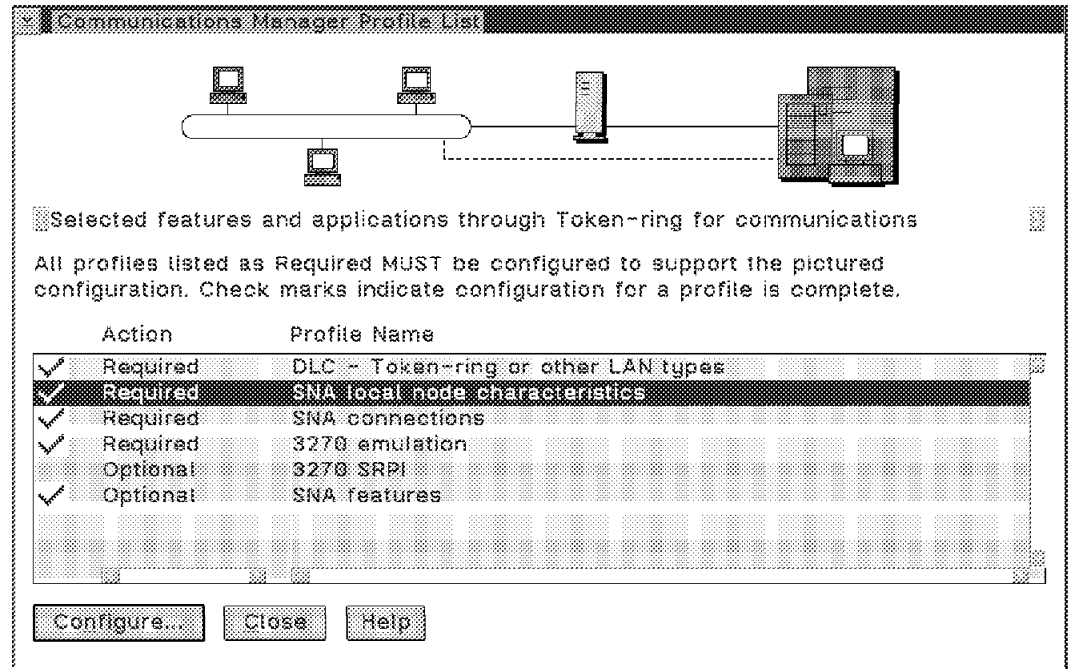


Figure 180. CM/2 Profile List - Selecting SNA Node Characteristics

Press the **Configure...** button on the previous window and you will be presented with the actual Local Node Characteristics window as shown in Figure 181.

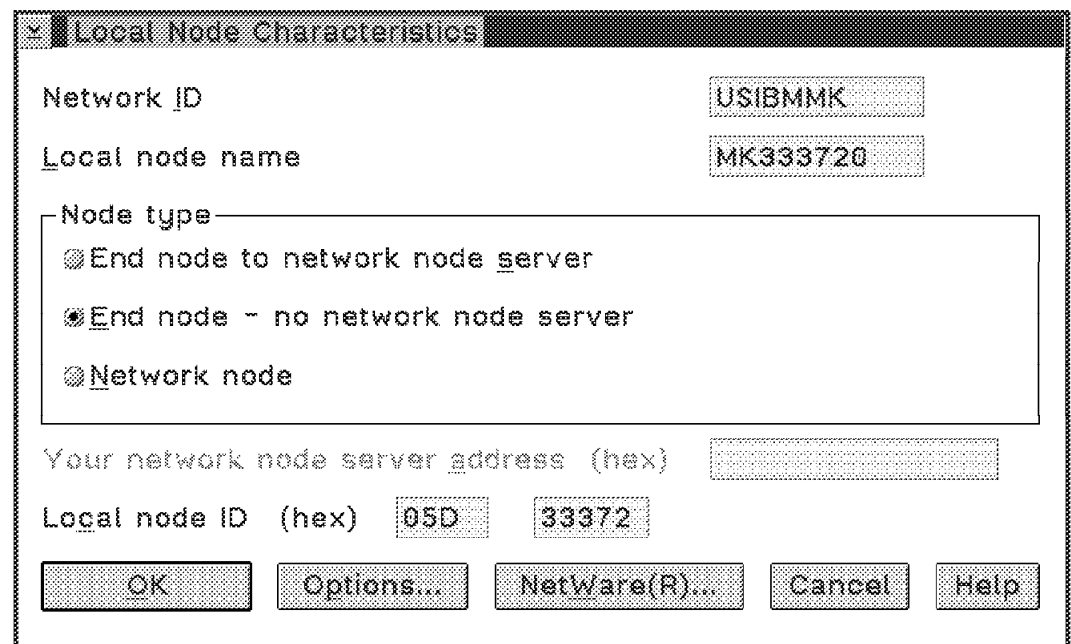


Figure 181. Setting SNA Local Node Characteristics

In the *Local Node Characteristics* window, we entered the following parameters:

- Enter USIBMMK in the *Network ID* field. This is the SNA NETID in which your node resides. When used with the local node name, it will become a fully qualified (FQ) Control Point (CP) name to uniquely identify your node in an interconnected network environment.

You can get the value of this field by contacting your network administrator or by using the NetView for MVS/ESA LISTVAR command.

- Enter MK333720 in the *Local node name* field. This is your CP name that was set up by your VTAM administrator. It will define a Logical Unit (LU name) that will be used for your APPC communication. Check Appendix B.2, "Host VTAM Definitions" on page 292 for an illustration of how this local node name was set up.
- Click on the **End Node - no network node server** radio button.
- Enter 05D 33372 in the Local node ID field. You will have to obtain these values from the SNA VTAM administrator for your installation. Check Appendix B.2, "Host VTAM Definitions" on page 292 for the entries IDBLK=05D and IDNUM=33372 to see where these came from.

We wanted to have an alias name for our local node, so we clicked on the **Options...** button and got the *Local Node Options* window as shown in Figure 182.

Figure 182. Setting SNA Local Node Options

In the preceding window, we entered our alias name as *WTR33372* (you can choose any name you want), clicked on the **Activate Attach Manager at start up** check box, and clicked on the **OK** button. This completed our SNA local node characteristics definition.

9.1.1.3 SNA Connections

We can now move to the next item for configuration which is the SNA connections. We click on the **SNA connections** line item as shown in Figure 183 on page 197.

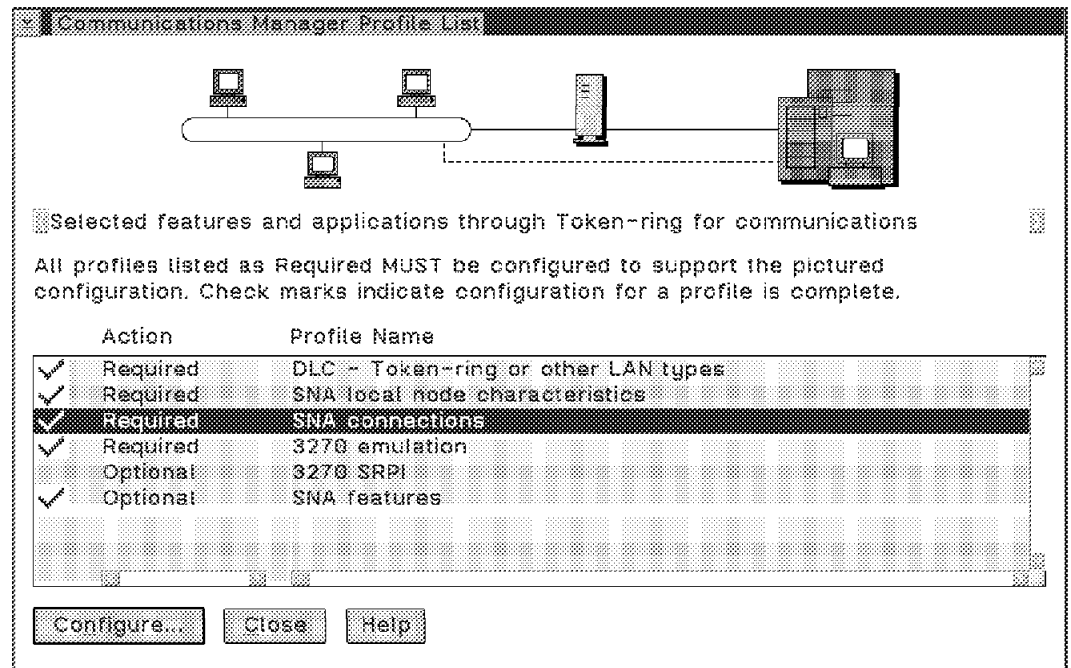


Figure 183. CM/2 Profile List - Selecting SNA Connections

After you click on the **Configure...** button on the previous window you will be presented with the actual SNA Connections List window as shown in Figure 184.

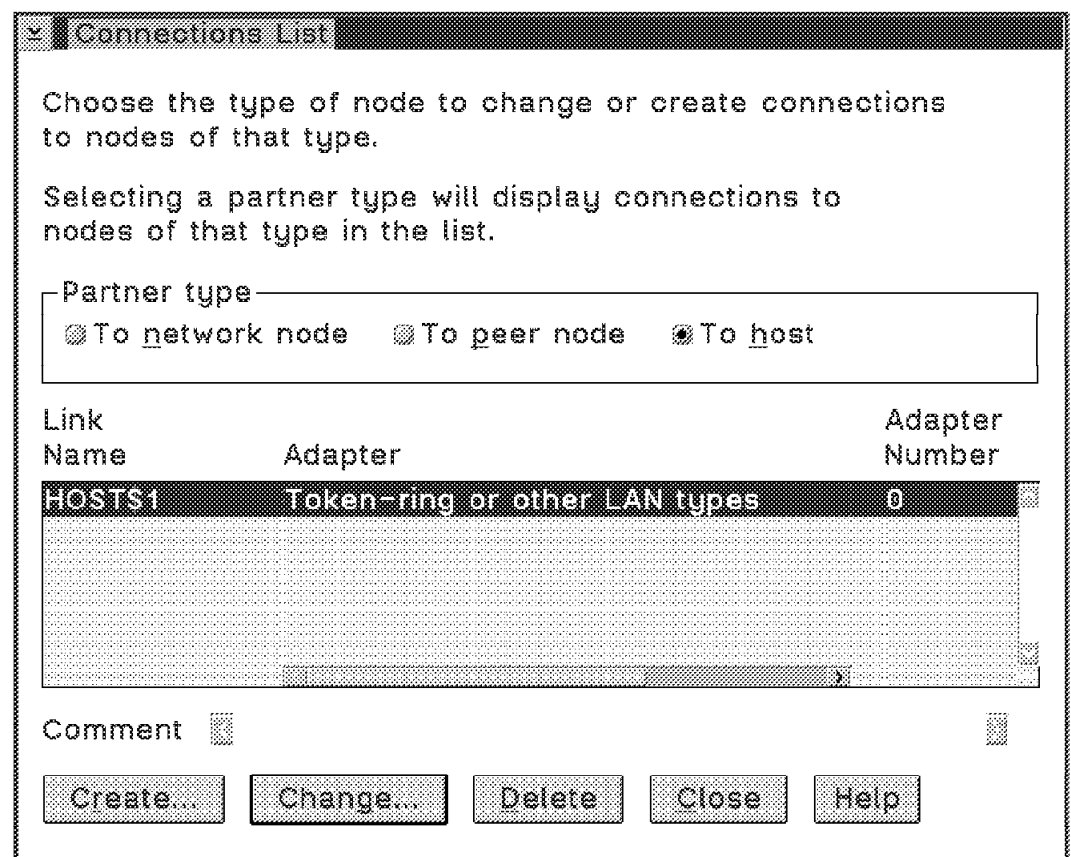


Figure 184. Communications Manager/2 Host Connections List

Note: There is a line item for *HOST\$1* already in the window because we had previously created this link. If you are setting this up for the first time, then there would be no Link Names defined and you would click on **Create...** to set up a new one. Also note that the To host button was selected.

We chose the Change... button to show you what parameters we entered to set up this host link. The first thing you see will be the *Adapter List* window as shown in Figure 185.

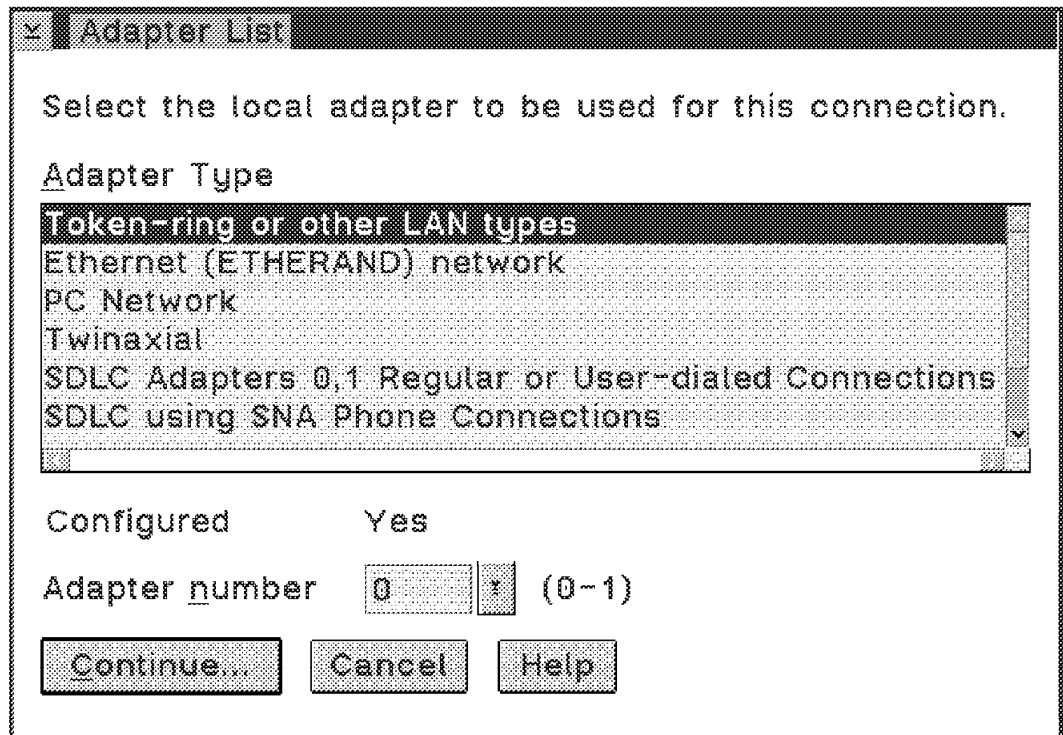


Figure 185. SNA Connections - Adapter Selection List

In the *Adapter List* window, we selected the line item for **Token-ring or other LAN types** and then clicked on the **Continue...** button. This presented us with the *Connection to a Host* window as shown in Figure 186 on page 199.

Connection to a Host

Link name: ☒ Activate at startup

Local PU name: ☐ APPN support

Node ID (hex):

LAN destination address (hex): Address format: Remote SAP (hex):

Adjacent node ID (hex):

Partner network ID:

Partner node name: (Required for partner LU definition)

☒ Use this host connection as your focal point support

Optional comment:

OK Define Partner LUs... Cancel Help

Figure 186. SNA Connections - Define Link Names and Destination Address

In the *Connection to a Host* window, we set up the following fields:

- Enter HOST\$1 in the *Link name* field. You can enter any link name that you like in this field.
- Make sure that the *Node ID (hex)* field has the same values that were entered in the *SNA Local Node Characteristics* panel as shown in Figure 181 on page 195.
- Make sure that the *LAN destination address* field contains the MAC address or Locally Administered Address (LAA) of the IBM 3745 Communications Controller that is token-ring connected to your backbone LAN and provides the gateway to your host machine. Our *Address format* is token-ring and the *Remote SAP* should be 04.
- Enter USIBMMK in the *Partner network ID* field. This link will get to the machine where we normally log on to. From there, the network will resolve the connection to the USIBMRA machine that we will set up as the true APPC partner in Figure 187 on page 200.
- Enter MK34 in the *Partner node name* field. The concatenation of partner network ID and partner node name will give you USIBMMK.MK34. Please refer to Appendix B.1, "CM/2 Configuration File" on page 291 to see the result of this definition which is to add a fully qualified adjacent CP Name (FQ_ADJACENT_CP_NAME(USIBMMK.MK34)) to our DEFINE_LOGICAL_LINK section.

This name will be referenced by the *DEFINE_PARTNER_LU_LOCATION* statement (when we create a partner LU) to indicate to CM/2 that our partner LU (USIBMRA.RAPAN) is located somewhere on the link that we defined.

- Click on the check box for *Use this host connection as your focal point support*.

- Click on the **Define Partner LUs...** button to get the Partner LUs window as shown in Figure 187 on page 200.

To add a Partner LU, enter the LU name, alias, and comment. Then select Add.

To change a Partner LU, select an LU from the list, change the LU name, alias, and/or comment fields and select Change.

To delete a Partner LU, select an LU from the list and select Delete.

Network ID:

LU name:

Alias:

Dependent partner LU

☐ Partner LU is dependent

Uninterpreted name:

LU name	Alias
USIBMRA.RAPAN	NETVIEW

Delete

Optional comment:

Add Change

OK Cancel Help

Figure 187. Define the Host NetView Application as Our APPC Partner LU

In Figure 187, you can see that our definitions have already been set up. To define our host NetView application as our partner LU, we typed in the following:

- USIBMRA as the *Network ID*.
- RAPAN as the *LU name*. Please refer to Appendix B.1, "CM/2 Configuration File" on page 291 to see what is defined in the *DEFINE_PARTNER_LU_LOCATION* section.

The *DEFINE_PARTNER_LU_LOCATION* section shows that USIBMMA.MK34 is our local owning CP name and that USIBMRA.RAPAN is the actual APPC partner. This will direct CM/2 to send a BIND for the requested partner LU on the link referenced in the *DEFINE_PARTNER_LU_LOCATION* statement.

- NETVIEW as the *Alias*. You can enter anything that you want for an alias name.

Once you click on OK, you are finished setting up your SNA connections and you can go back to the Communications Manager/2 Profile List window as shown in Figure 188 on page 201.

Note: We skipped over the *3270 Emulation* configuration section. If you are setting up *3270 Emulation* for the first time, you just need to enter the number of 3270 sessions that you want when Communications Manager/2 starts up. You can use one of these sessions to access host NetView on your NetView for OS/2 managing system, but normally the host NetView console would be on a separate display. There are a few 3270-type options that you can customize, but they are not related to our scenario here.

We can now set up the optional SNA Features.

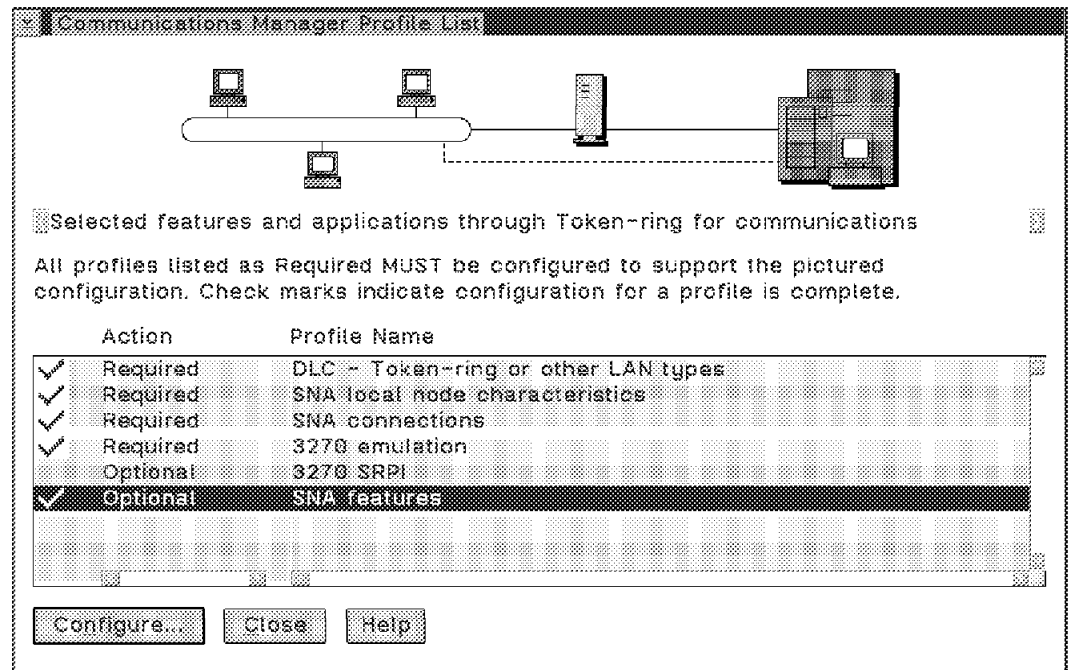


Figure 188. Communications Manager/2 Profile List - Selecting SNA Features

In the *Profile List* window, we select the line item for SNA features and then click on the **Continue...** button. This will present us with the *SNA Features List* window as shown in Figure 189 on page 202.

Note that the *Local LUs* line item has been selected for you and that nothing appears in the *Definition* scroll box. You do not need to create a local LU here because it would be the second LU on your node. You can simply use the Control Point LU (USIBMMK.MK333720) that you set up earlier. When you start up Communications Manager/2, you will be initiating an implicit focal point relationship with the host NetView application by establishing an LU 6.2 session between your Control Point LU and the NetView Partner LU (USIBMRA.RAPAN). Please see 9.1.2, "Defining an Implicit Focal Point Relationship" on page 203 to find out exactly what you need to enter in your NDF file to accomplish this.

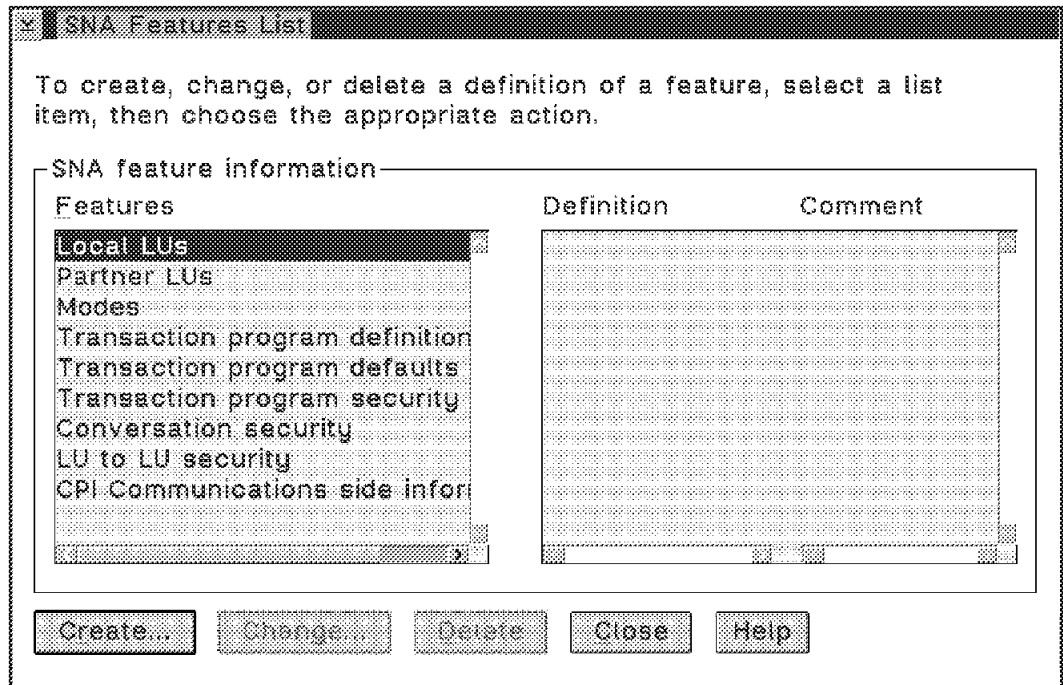


Figure 189. CM/2 Configuration - SNA Features List - Local LUs

If you click on the **Partner LUs** line item as shown in Figure 190, you will notice that the definition for *NETVIEW* comes up.

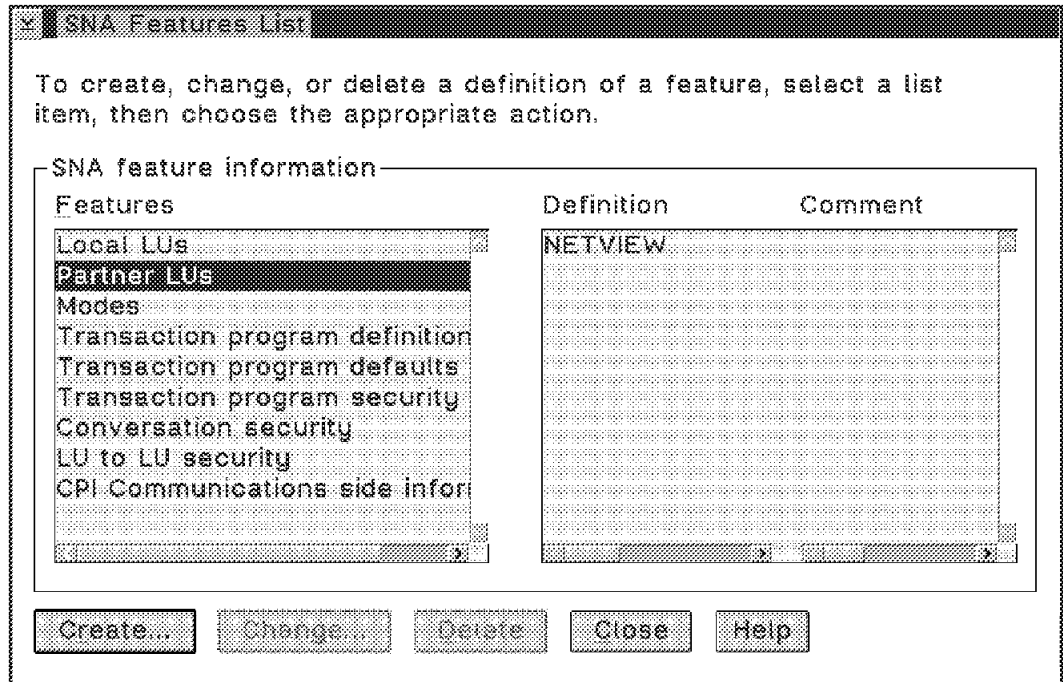


Figure 190. CM/2 Configuration - SNA Features List - Partner LUs

We defined this *NETVIEW* alias in the SNA Connections section when we set up our partner LUs as shown in Figure 187 on page 200.

If you click on the **Modes** line item as shown in Figure 191 on page 203, you will see a list of valid modes that have been set up on the system. These are the

standard IBM-supplied modes that you can use without having to create new ones if they meet your requirements. If you want to select one for your use, we found that the #INTER definition worked fine in this environment.

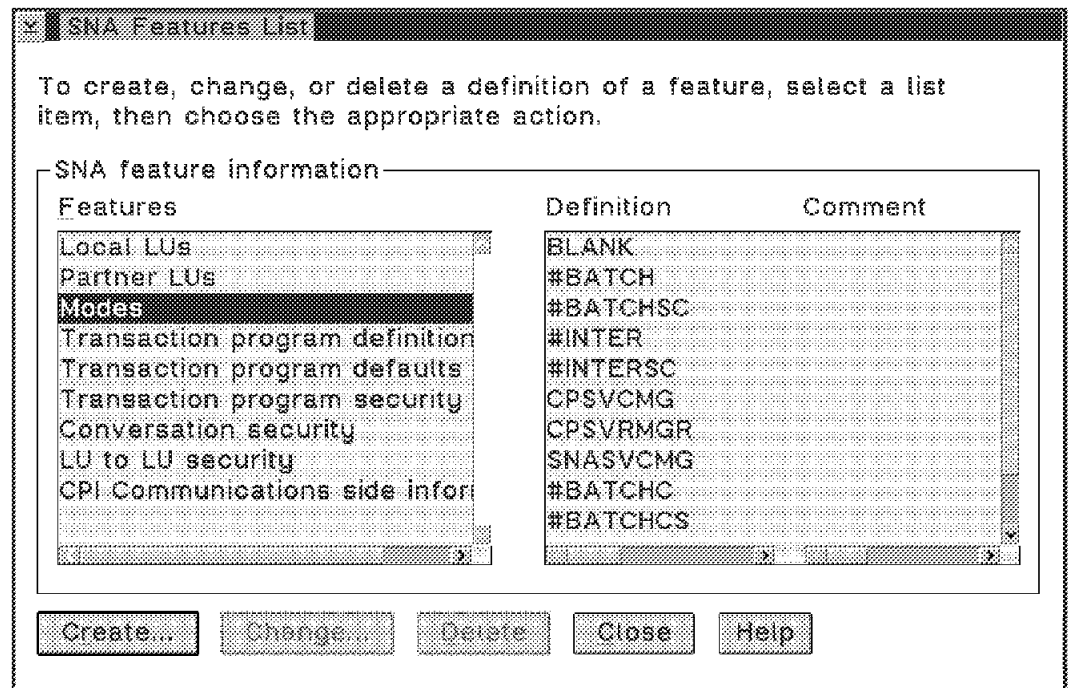


Figure 191. CM/2 Configuration - SNA Features List - Modes

Since we made no changes, just click on the **Close** button to return to the Communications Manager/2 Profile List window.

Click on the **Close** button to return to the Communications Manager/2 Configuration Definition window.

Click on the **Close** button to exit Communications Manager/2. The configuration process will automatically verify your configuration, and if you have CM/2 started, it will ask you if you want to dynamically update your SNA resources. You can do this but you still have to edit your NDF file to include implicit focal point support. Therefore, you will have to stop and restart Communications Manager/2 after making the changes described in 9.1.2, "Defining an Implicit Focal Point Relationship."

9.1.2 Defining an Implicit Focal Point Relationship

In order for your system to initiate an implicit relationship with a remote focal point (host NetView), you have to manually add the following lines to your *.NDF file:

```
DEFINE_REMOTE_FOCAL_POINT SNA_DEFINED_MS_CATEGORY(X'23',031)
                           DESCRIPTION(ALERT CATEGORY)
                           FQ_PRIMARY_FP_NAME(USIBMRA.RAPAN );
```

When Communications Manager/2 starts on your machine, you will be initiating an implicit focal point relationship with NetView on the host, instead of having host NetView change its focal point definitions to include your station. Please refer to Appendix B.1, "CM/2 Configuration File" on page 291 to see where we inserted the above lines.

You will now have to stop and restart Communications Manager/2.

9.1.3 Setting Up NetView on the Host

The only thing that you will have to make sure of from a host NetView perspective, is that the task DSI6DST is active. This started task will enable LU 6.2 transports. To start this task, enter the following command at a NetView operator Command Facility command line:

```
START TASK=DSI6DST
```

9.1.4 Testing Your CM/2 to Host NetView Link

Once you have stopped and restarted CM/2 using your new *.NDF file and DSI6DST is started on host NetView, you should have an active LU 6.2 session between your NetView for OS/2 managing station and NetView on the MVS host. To monitor the status of your resources, go to your Communications Manager/2 main window of icons and click on the **Subsystem Management** icon as shown in Figure 192.

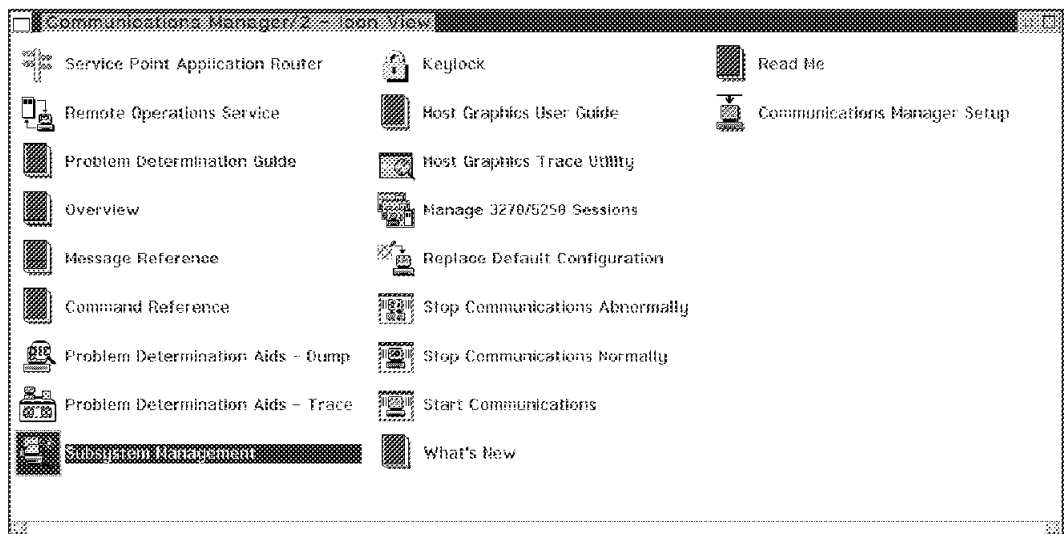


Figure 192. Initiating Subsystem Management to Test Your Configuration

You will be presented with the *Subsystem Management* window. The following services should display a status of "Started":

- APPC attach manager
- Communications Manager/2 kernel
- SNA subsystem

Click on the **SNA Subsystem** line item as shown in Figure 193 on page 205.

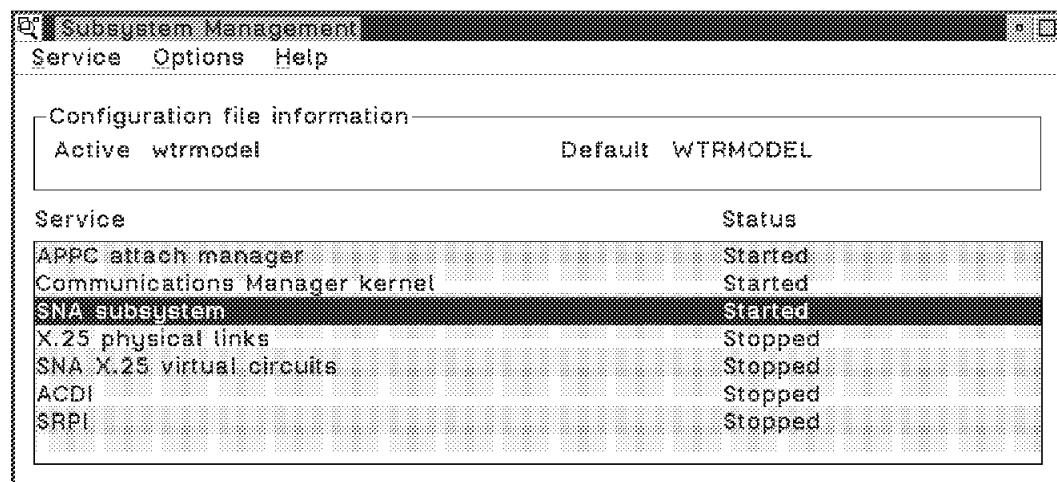


Figure 193. Subsystem Management - All Subsystems Started

On the *SNA Subsystem* window, select the LU 6.2 sessions line item as shown in Figure 194.

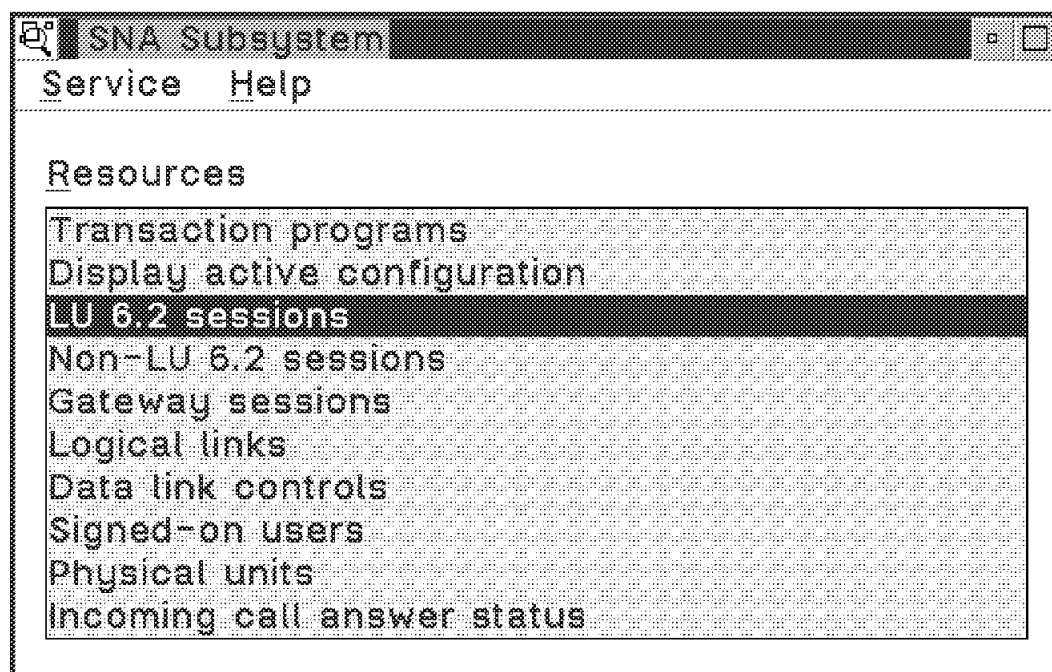


Figure 194. Selecting LU 6.2 Sessions from SNA Subsystems

If you selected to have your LU 6.2 session initiated at CM/2 startup, you should already have an active session as indicated by having a line already in your *LU 6.2 Sessions* window as shown in Figure 195 on page 206.

The screenshot shows a window titled "LU 6.2 Sessions" with a menu bar containing "Session", "Establish", "Options", and "Help". Below the menu bar is a table with the following data:

Session	Establish	Options	Help
WTR33372	NETVIEW	USIBMRA.RAPAN	SNASVCMG 1

Figure 195. Successful LU 6.2 Session Establishment

Notice that CM/2 displays the local alias name (WTR33372), my partner's alias name (NETVIEW), my partner's fully qualified LU Name (USIBMRA.RAPAN), the mode it used (SNASVCMG) and how many sessions of that mode are active.

9.1.5 Using and Changing Alert Filters

Before you start the Host Connection program, you should think about how you are going to limit the number of alerts being sent to host NetView. You probably do not want to send every trap that gets displayed in your Event Displayer window.

The default filter that gets used when you start the Host Connection program will forward all SNMP trap types, from type 0 through type 6. The *IBM NetView for OS/2 User's Guide*, SC31-8099 covers the topic of creating and changing event filters. It describes in detail how to create your RuleContent, what operators are allowed, and what parameters you can use to limit flow of alerts based on Class, IP Address, SNMP Trap Number, and SNMP Specific Trap Values. This document will show how we created our own filter and then started the Host Connection program using that filter.

9.1.5.1 Modifying the Supplied Alert Filter File

The default alert filter file supplied with NetView for OS/2 is found in the following file:

```
\anv2\etc\tralert.flr
```

Here are the contents of the file before we started editing it:

```

RuleName=Trap_to_Alert_Default_Filter
RuleDescription=Default filter for trap-to-alert conv. on startup of tralertd
RuleContent=(SNMP_TRAP=0 || SNMP_TRAP=1 ||
SNMP_TRAP=2 || SNMP_TRAP=3 || SNMP_TRAP=4 || SNMP_TRAP=5 || SNMP_TRAP=6)

RuleName=Trap_to_Alert_Filter_Sample
RuleDescription=Sample filter for trap-to-alert
RuleContent=((CLASS=1.3.6.1.4.1.2.6.3 && (SNMP_TRAP=0 || SNMP_TRAP=1 ||
|| SNMP_TRAP=2 || SNMP_TRAP=3 || SNMP_TRAP=4 || SNMP_TRAP=5 ||
SNMP_SPECIFIC=58720256 || SNMP_SPECIFIC=58720257 || SNMP_SPECIFIC=58720258 ||
SNMP_SPECIFIC=58720259 || SNMP_SPECIFIC=58720260 ||
SNMP_SPECIFIC=58720261 || SNMP_SPECIFIC=58720262 || SNMP_SPECIFIC=58720263 || SNMP_SPECIFIC=58720264
|| SNMP_SPECIFIC=58851330 || SNMP_SPECIFIC=58916864 || SNMP_SPECIFIC=58916865 ||
SNMP_SPECIFIC=58916866 || SNMP_SPECIFIC=58916867 || SNMP_SPECIFIC=58916868 || SNMP_SPECIFIC=58916869))
|| (CLASS=1.3.6.1.4.1.2.6.4)
|| (CLASS=1.3.6.1.4.1.2.6.2))

RuleName=Receive_from_6611_router_sample
RuleDescription=Receive enterprise-specific events from 6611 router
RuleContent=((CLASS=1.3.6.1.4.1.2.6.2 && (SNMP_SPECIFIC=1 || SNMP_SPECIFIC=2 ||
SNMP_SPECIFIC=3 || SNMP_SPECIFIC=4 || SNMP_SPECIFIC=5 || SNMP_SPECIFIC=6 ||
SNMP_SPECIFIC=7 || SNMP_SPECIFIC=8 || SNMP_SPECIFIC=9 || SNMP_SPECIFIC=10 ||
SNMP_SPECIFIC=11 || SNMP_SPECIFIC=12 || SNMP_SPECIFIC=13 ||
SNMP_SPECIFIC=14 || SNMP_SPECIFIC=15 || SNMP_SPECIFIC=16 || SNMP_SPECIFIC=17 ||
SNMP_SPECIFIC=18 || SNMP_SPECIFIC=19 || SNMP_SPECIFIC=20))) &&
(IP_ADDR=9.67.14.24 || IP_ADDR=haydn || IP_ADDR=9.67.8.4)

RuleName=Trap_to_Alert_Threshold_sample
RuleDescription=convert trap to an alert
RuleContent=PRESENT=SNMP_TRAP

```

Figure 196. Contents of TRALERT.FLT File

Look at the specification for the RuleName for the Trap_to_Alert_Default_Filter. It will allow all SNMP traps to pass to NetView on the host (trap types 0 through 6).

We are going to add our own rule to allow only LAN Server and LAN Requester traps to be forwarded to host NetView. If we use the MIB Browser (as explained in 3.3, “MIB Browser” on page 62), we can find the MIB Variable entries for both LAN Server and Requester in the *private.enterprises.ibm.ibmProd* tree. If we use the MIB Browser describe function on these two variables we will find their matching ASN.1 dot notation CLASS identifier as shown in Table 2.

Table 2. LAN Server and LAN Requester CLASS Identifiers	
MIB Variable Name	CLASS Identifier
os2LS	1.3.6.1.4.1.2.6.57
os2LReq	1.3.6.1.4.1.2.6.58

To build our own filter/rule for allowing traps from only OS/2 LAN Server and LAN Requester to flow, we added the following lines to the TRALERT.FLT file as shown in Figure 197:

```

RuleName=LAN_Server_Requester_Only_Filter
RuleDescription=Sample filter for LAN Server and Requester
RuleContent=((CLASS=1.3.6.1.4.1.2.6.57 && (SNMP_SPECIFIC=1))
|| (CLASS=1.3.6.1.4.1.2.6.58 && (SNMP_SPECIFIC=1)))

```

Figure 197. RuleContent for LAN Server/Requester Filter

Warning

RuleContent parsing is done by the North Star parser which was ported from NetView for AIX. This parser is very strict. When you want to further restrict a Class by an SNMP_TRAP type or an SNMP_SPECIFIC value, ensure that you do not have a closing bracket) after the Class number.

9.1.5.2 Modifying the Alert Configuration File

The alert configuration file contains:

```
d:\anv2\etc\tralert.flr Trap_to_Alert_Default_Filter
```

The alert configuration file should have the following name:

```
\anv2\etc\tralert.cfg
```

Here is the contents of the alert configuration file to point to our filter for LAN Server/Requester only alerts:

```
D:\anv2\etc\tralert.flr LAN_Server_Requester_Only_Filter
```

Figure 198. Contents of TRALERT.CFG File

Notice that the TRALERT.CFG file contains two parameters:

1. The fully qualified file name of the file containing your filter.
2. The *RuleName* of the event filter that you want to use.

Note: The TRALERT.CFG file can contain only *one* line. You cannot start the Host Connection program if you have more than one filter.

9.1.6 Starting the NetView for OS/2 Host Connection Program

Before starting the Host Connection program, please ensure that the NetView for OS/2 Process Control Manager and CM/2 are running and that your APPC connection has been established.

You can start the NetView for OS/2 Host connection program by clicking on the Host Connection icon on the NetView for OS/2 Icon View window as shown in Figure 199.

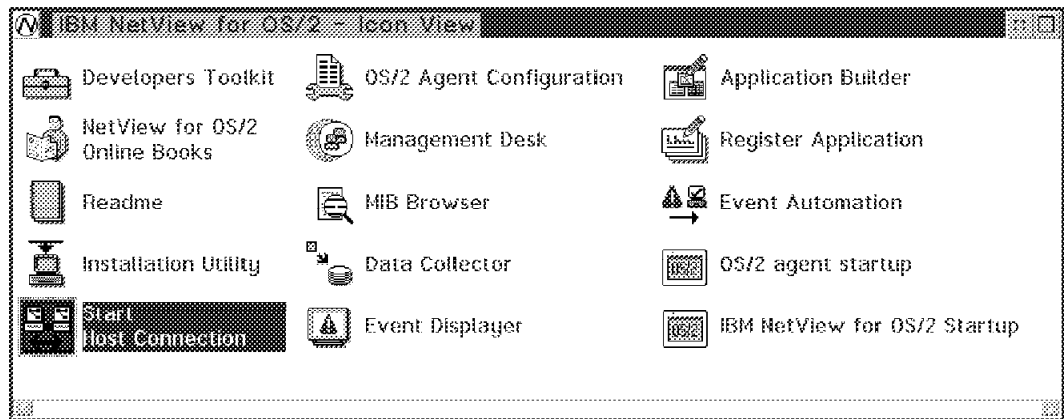


Figure 199. Starting the NetView for OS/2 Host Connection Program

You can also start the Host Connection program from an OS/2 command prompt. The name of the program is *TRALERTD.EXE* and is found in the *\anv2\bin* directory. Therefore, to start the TRALERTD daemon, enter the following at a command prompt:

START TRALERTD

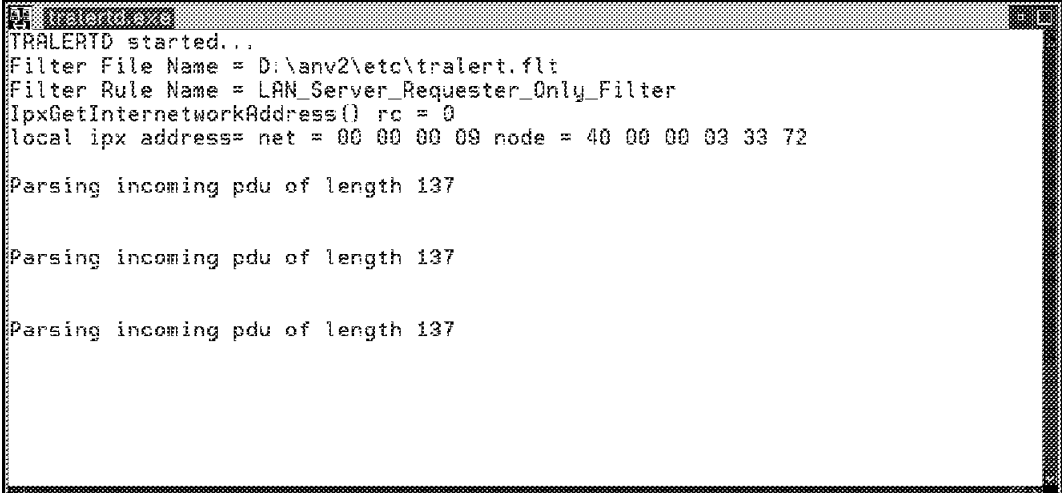
No matter which way you start TRALERTD, it will always use the current contents of the TRALERT.CFG file to initiate the RuleName that you have defined. If no TRALERT.CFG file exists, it will use the *Trap_to_Alert_Default_Filter* in the TRALERT.FLT file.

Warning

The RuleNames in the TRALERT.FLT file are all case sensitive.

Once you start the TRALERTD daemon, it will open a separate OS/2 window that will show it has been started, which filter file it is using, and what RuleName it is using.

When we started TRALERTD using the RuleName that we created for the *LAN_Server_Requester_Only_Filter*, the startup window appears as shown in Figure 200.



```
TRALERTD started...
Filter File Name = D:\anv2\etc\tralert.flt
Filter Rule Name = LAN_Server_Requester_Only_Filter
IpxGetInternetworkAddress() rc = 0
local ipx address= net = 00 00 00 09 node = 40 00 00 03 33 72

Parsing incoming pdu of length 137

Parsing incoming pdu of length 137

Parsing incoming pdu of length 137
```

Figure 200. Starting TRALERTD with our LAN Server/Requester RuleContent

When we started TRALERTD using the *Trap_to_Alert_Default_Filter*, the startup window appears as shown in Figure 201 on page 210.

```

TRALERTD started...
Filter File Name = D:\any2\etc\tralert.flt
Filter Rule Name = Trap_to_Alert_Default_Filter
IpxGetInterNetworkAddress() rc = 0
local ipx address= net = 00 00 00 09 node = 40 00 00 03 33 72

Parsing incoming pdu of length 120

Parsing incoming pdu of length 120

Parsing incoming pdu of length 120

```

Figure 201. Successful Start of the Host Connection Program Using Defaults

Every time a trap comes into the Event Displayer, it also comes to the TRALERTD.EXE window and a message is displayed saying that it is *Parsing incoming pdu of length n*, where *n* is the length in bytes of the trap message. If this trap message meets all alert filtering criteria, it is then passed up to host NetView using the LU 6.2 link.

If we look at the Event Displayer window as shown in Figure 202, we see that 5 traps showed up.

All Events			
Time	Node	Generic Specific	Description
Aug 25 21:08:27 1994 9.24.104.68		4	Authentication failure Trap: Inco
Aug 25 21:08:28 1994 9.24.104.68		4	Authentication failure Trap: Inco
Aug 25 21:08:29 1994 9.24.104.68		4	Authentication failure Trap: Inco
Aug 25 21:13:27 1994 9.24.104.55		6 1	Enterprise: { iso.org.dod.internet.private.ed.od.os2Req.os2LANRequester.rdpr
Aug 25 21:13:28 1994 9.24.104.55		6 1	Enterprise: { iso.org.dod.internet.private.ed.od.os2Req.os2LANRequester.trupe

Figure 202. SNMP Traps which Will Be Forwarded to Host NetView

The first three traps came in at approximately 21:08 and the last two traps came in at approximately 21:13. Since we were using the *Trap_to_Alert_Default_Filter*, all SNMP traps were being forwarded to host NetView. We can see them on the top five lines of the following NetView Hardware Monitor - Alerts Dynamic screen as shown in Figure 203 on page 211. Notice the alert TYPE is *SNMP*.

```

A - A - 3270 Emulator
File Edit Transfer Settings Keyboard Help
NETVIEW SESSION DOMAIN: RAPAN WTWKS1 08/25/94 21:13:01
NPDA-008 * ALERTS-DYNAMIC *

DOMAIN RESNAME TYPE TIME ALERT DESCRIPTION:PROBABLE CAUSE
RAPAN 9_24_104 SNMP 21:13 SNMP RESOURCE PROBLEM:UNDETERMINED
RAPAN 9_24_104 SNMP 21:13 SNMP RESOURCE PROBLEM:UNDETERMINED
RAPAN 9_24_104 SNMP 21:08 POSS UNAUTH ACC ATTEMPTED:SECURITY PROBLEM
RAPAN 9_24_104 SNMP 21:08 POSS UNAUTH ACC ATTEMPTED:SECURITY PROBLEM
RAPAN 9_24_104 SNMP 21:08 POSS UNAUTH ACC ATTEMPTED:SECURITY PROBLEM
RAPAN RA6003CP*DEV 20:49 PROBLEM RESOLVED:REMOTE NODE
RAPAN RA6003CP*DEV 20:49 PROBLEM RESOLVED:COMMUNICATIONS INTERFACE
RAPAN RA7NCPV COMC 20:40 CONFIG/CUSTOMIZATION ERR:CONFIGURATION
RAPAN RA7NCPV COMC 20:40 CONFIG/CUSTOMIZATION ERR:CONFIGURATION
RAPAN RA7NCPV COMC 20:39 CONFIG/CUSTOMIZATION ERR:CONFIGURATION
RAPAN RS60003 DEV 20:36 OPERATOR NOTIFICATION:NETWORK OPERATOR
RAPAN RA6003CP*DEV 20:36 PROBLEM RESOLVED:REMOTE NODE
RAPAN RA6003CP*DEV 20:36 PROBLEM RESOLVED:COMMUNICATIONS INTERFACE
RAPAN RA7NCPV COMC 20:34 CONFIG/CUSTOMIZATION ERR:CONFIGURATION
RAPAN RA7NCPV COMC 20:33 CONFIG/CUSTOMIZATION ERR:CONFIGURATION
RAPAN RA7NCPV COMC 20:33 CONFIG/CUSTOMIZATION ERR:CONFIGURATION
RAPAN RS60003 DEV 20:24 OPERATOR NOTIFICATION:NETWORK OPERATOR
RAPAN RS60003 DEV 20:24 OPERATOR NOTIFICATION:NETWORK OPERATOR
RAPAN RS60003 DEV 20:20 OPERATOR NOTIFICATION:NETWORK OPERATOR
RAPAN RS60003 DEV 20:20 OPERATOR NOTIFICATION:NETWORK OPERATOR
RAPAN RA6003CP*DEV 20:00 PROBLEM RESOLVED:REMOTE NODE
RAPAN RA6003CP*DEV 20:00 PROBLEM RESOLVED:COMMUNICATIONS INTERFACE

DEPRESS ENTER KEY TO VIEW ALERTS-STATIC

??
CMD==>

```

Figure 203. NetView for MVS - Hardware Monitor Alerts-Dynamic Screen

9.1.7 Using NetView for MVS to Monitor SNMP Devices

In this section, we will show how to use NetView to monitor an OS/2 SNMP device for authentication failures. To display alerts using host NetView, follow these steps:

1. Log on to your local host NetView console.
2. Type the *NPDA* command at the first command prompt.
This will put you in the Hardware Monitor application.
3. Enter a *1* to access the Alerts-Dyanmic funtion of NetView. You can also get to this screen directly from any NetView command prompt by entering *NPDA ALD*.

You will get a NetView screen similar to the one shown in Figure 204 on page 212.

```

A 3270 Emulator
File Edit Transfer Settings Keyboard Help
N E T V I E W          SESSION DOMAIN: RAPAN      WTKSH1    08/25/94 21:34:53
NPDA-30A                * ALERTS-DYNAMIC *

DOMAIN RESNAME TYPE TIME ALERT DESCRIPTION:PROBABLE CAUSE
RAPAN RA7NCPV COMC 21:34 CONFIG/CUSTOMIZATION ERR:CONFIGURATION
RAPAN RA7NCPV COMC 21:34 CONFIG/CUSTOMIZATION ERR:CONFIGURATION
RAPAN RA7NCPV COMC 21:34 CONFIG/CUSTOMIZATION ERR:CONFIGURATION
RAPAN 9_24_104 SNMP 21:20 SNMP RESOURCE PROBLEM:UNDETERMINED
RAPAN 9_24_104 SNMP 21:20 SNMP RESOURCE PROBLEM:UNDETERMINED
RAPAN NV2MGR1_ SNMP 21:18 POSS UNAUTH ACC ATTEMPTED:SECURITY PROBLEM
RAPAN NV2MGR1_ SNMP 21:18 POSS UNAUTH ACC ATTEMPTED:SECURITY PROBLEM
RAPAN NV2MGR1_ SNMP 21:18 POSS UNAUTH ACC ATTEMPTED:SECURITY PROBLEM
RAPAN 9_24_104 SNMP 21:13 SNMP RESOURCE PROBLEM:UNDETERMINED
RAPAN 9_24_104 SNMP 21:13 SNMP RESOURCE PROBLEM:UNDETERMINED
RAPAN 9_24_104 SNMP 21:08 POSS UNAUTH ACC ATTEMPTED:SECURITY PROBLEM
RAPAN 9_24_104 SNMP 21:08 POSS UNAUTH ACC ATTEMPTED:SECURITY PROBLEM
RAPAN 9_24_104 SNMP 21:08 POSS UNAUTH ACC ATTEMPTED:SECURITY PROBLEM
RAPAN RA6003CP*DEV 20:49 PROBLEM RESOLVED:REMOTE NODE
RAPAN RA6003CP*DEV 20:49 PROBLEM RESOLVED:COMMUNICATIONS INTERFACE
RAPAN RA7NCPV COMC 20:40 CONFIG/CUSTOMIZATION ERR:CONFIGURATION
RAPAN RA7NCPV COMC 20:40 CONFIG/CUSTOMIZATION ERR:CONFIGURATION
RAPAN RA7NCPV COMC 20:39 CONFIG/CUSTOMIZATION ERR:CONFIGURATION
RAPAN RS60003 DEV 20:36 OPERATOR NOTIFICATION:NETWORK OPERATOR
RAPAN RA6003CP*DEV 20:36 PROBLEM RESOLVED:REMOTE NODE
RAPAN RA6003CP*DEV 20:36 PROBLEM RESOLVED:COMMUNICATIONS INTERFACE
RAPAN RA7NCPV COMC 20:34 CONFIG/CUSTOMIZATION ERR:CONFIGURATION
RAPAN RA7NCPV COMC 20:33 CONFIG/CUSTOMIZATION ERR:CONFIGURATION

DEPRESS ENTER KEY TO VIEW ALERTS-STATIC

???
CMD==>

```

Figure 204. Sample Host NetView Alert Screen Showing SNMP Alerts

If you now want to research one of these alerts for more detail, press the *Enter* key, and the screen will change from Alerts-Dynamic to Alerts-Static (the screen content will not change). Selection numbers will be placed along the left side of each alert entry as shown in Figure 205 on page 213.


```

A - A - 3270 Emulator
File Edit Transfer Settings Keyboard Help
N E T V I E W          SESSION DOMAIN: RAPAN      WTKSH1    08/25/94 21:39:22
NPDAR-308              * ALERTS-STATIC *

SEL# DOMAIN RESNAME TYPE TIME  ALERT DESCRIPTION: PROBABLE CAUSE
( 1) RAPAN RA7NCPV  COMC 21:34 CONFIG/CUSTOMIZATION ERR: CONFIGURATION
( 2) RAPAN RA7NCPV  COMC 21:34 CONFIG/CUSTOMIZATION ERR: CONFIGURATION
( 3) RAPAN RA7NCPV  COMC 21:34 CONFIG/CUSTOMIZATION ERR: CONFIGURATION
( 4) RAPAN 9_24_104 SNMP 21:20 SNMP RESOURCE PROBLEM: UNDETERMINED
( 5) RAPAN 9_24_104 SNMP 21:20 SNMP RESOURCE PROBLEM: UNDETERMINED
( 6) RAPAN NV2MGR1_ SNMP 21:18 POSS UNAUTH ACC ATTEMPTED: SECURITY PROBLEM
( 7) RAPAN NV2MGR1_ SNMP 21:18 POSS UNAUTH ACC ATTEMPTED: SECURITY PROBLEM
( 8) RAPAN NV2MGR1_ SNMP 21:18 POSS UNAUTH ACC ATTEMPTED: SECURITY PROBLEM
( 9) RAPAN 9_24_104 SNMP 21:13 SNMP RESOURCE PROBLEM: UNDETERMINED
(10) RAPAN 9_24_104 SNMP 21:13 SNMP RESOURCE PROBLEM: UNDETERMINED
(11) RAPAN 9_24_104 SNMP 21:08 POSS UNAUTH ACC ATTEMPTED: SECURITY PROBLEM
(12) RAPAN 9_24_104 SNMP 21:08 POSS UNAUTH ACC ATTEMPTED: SECURITY PROBLEM
(13) RAPAN 9_24_104 SNMP 21:08 POSS UNAUTH ACC ATTEMPTED: SECURITY PROBLEM
(14) RAPAN RA6003CP*DEV 20:49 PROBLEM RESOLVED: REMOTE NODE
(15) RAPAN RA6003CP*DEV 20:49 PROBLEM RESOLVED: COMMUNICATIONS INTERFACE
(16) RAPAN RA7NCPV  COMC 20:40 CONFIG/CUSTOMIZATION ERR: CONFIGURATION
(17) RAPAN RA7NCPV  COMC 20:40 CONFIG/CUSTOMIZATION ERR: CONFIGURATION
(18) RAPAN RA7NCPV  COMC 20:39 CONFIG/CUSTOMIZATION ERR: CONFIGURATION
(19) RAPAN RS60003  DEV  20:36 OPERATOR NOTIFICATION: NETWORK OPERATOR
(20) RAPAN RA6003CP*DEV 20:36 PROBLEM RESOLVED: REMOTE NODE
(21) RAPAN RA6003CP*DEV 20:36 PROBLEM RESOLVED: COMMUNICATIONS INTERFACE
(22) RAPAN RA7NCPV  COMC 20:34 CONFIG/CUSTOMIZATION ERR: CONFIGURATION
(23) RAPAN RA7NCPV  COMC 20:33 CONFIG/CUSTOMIZATION ERR: CONFIGURATION
DEPRESS ENTER KEY TO VIEW ALERTS-DYNAMIC OR ENTER A TO VIEW ALERTS-HISTORY
ENTER SEL# (ACTION), OR SEL# PLUS M (MOST RECENT), P (PROBLEM), DEL (DELETE)

???
CMD==> 6

```

Figure 205. Host NetView Alerts-Static Screen

You can now select an alert that you want more detail on. In the previous Alerts-Static screen, notice that we entered a 6 on the command line and pressed enter to get the Recommended Actions screen for that particular event as shown in Figure 206 on page 214. We can see that this particular event came from our NV2MGR1 machine and that the Alert Description is *POSS UNAUTH ACC ATTEMPTED: SECURITY PROBLEM*. We can now review the recommended actions.

```

A - A - 3270 Emulator
File Edit Transfer Settings Keyboard Help
N E T V I E W      SESSION DOMAIN: RAPAN      WTKSH1      08/25/94 21:41:00
NPDR-45A          * RECOMMENDED ACTION FOR SELECTED EVENT *      PAGE 1 OF 1
RAPAN            MK333720      UNKNOWN      NV2MGR1_
+-----+      +-----+      +-----+
DOMAIN          | CP      |---| LLC      |---| SNMP      |
+-----+      +-----+      +-----+

ACTIONS - I008 - PERFORM PROBLEM DETERMINATION PROCEDURE FOR ORIGINATOR
9.24.104.54
I588 - IF REQUIRED, QUERY CONTACT ID Mirek Iwachow AT LOCATION
NAME Raleigh bld 062 room L610 ABOUT COMPONENT ID
nv2mgr1.itso.ral.ibm.com

ENTER ST (MOST RECENT STATISTICS), DM (DETAIL MENU), OR D (EVENT DETAIL)

???
CMD==>

```

Figure 206. Host NetView - Recommended Actions for Selected Event Screen

In Figure 206, we can now see how important it is to enter meaningful and descriptive names for your SNMP devices when you are configuring them. The message tells us who to contact, at what location, and provides a descriptor field.

To get a more detailed display of this *possible unauthorized access attempt*, you can enter *DM* on the previous panel to get the *Event Detail Menu* screen as shown in Figure 207 on page 215.

```

A - A - 3270 Emulator
File Edit Transfer Settings Keyboard Help
N E T V I E W      SESSION DOMAIN: RAPAN   WTKSH1   08/25/94 21:42:12
NPDR-43R           * EVENT DETAIL MENU *
                                           PAGE 1 OF 1

RAPAN      MK333720      UNKNOWN      NV2MGR1_
+-----+ +-----+ +-----+
DOMAIN    | CP  | --- | LLC  | --- | SNMP |
+-----+ +-----+ +-----+

DATE/TIME: 08/25 21:18

SEL#  PRODUCES:
( 1)  EVENT DETAIL DISPLAY
( 2)  PRODUCT SET IDENTIFICATION DISPLAY
( 3)  HEXADECIMAL DISPLAY OF DATA RECORD

ENTER SEL# OR A (ACTION)

???
CMD==>

```

Figure 207. Host NetView - Event Detail Display Menu

On this Event Detail Menu, enter a 1 and press the Enter key. You will get an *Event Detail* screen that shows you additional information. Notice that this information is very similar to what you would see on the NetView for OS/2 Event Displayer window. For example, at a specific date and time, there was a possible unauthorized access attempted as notified by the SNMP Generic-Trap Number Authentication Failure. All these details are shown in Figure 208 on page 216.

```

A - A - 3270 Emulator
File Edit Transfer Settings Keyboard Help
N E T V I E W      SESSION DOMAIN: RAPAN      WTKSH1      08/25/94 21:43:05
NPDA-43S          * EVENT DETAIL *
PAGE 1 OF 2

RAPAN      MK333720      UNKNOWN      NV2MGR1_
+-----+
DOMAIN      | CP      |---| LLC      |---| SNMP      |
+-----+

DATE/TIME: RECORDED - 08/25 21:18      CREATED - 08/25/94 21:18:38

EVENT TYPE: IMPENDING PROBLEM

DESCRIPTION: POSSIBLE UNAUTHORIZED ACCESS ATTEMPTED

PROBABLE CAUSES:
  SECURITY PROBLEM

QUALIFIERS:
  1) ENTERPRISE 1.3.6.1.4.1.2.6.61
  2) SNMP GENERIC-TRAP NUMBER AUTHENTICATION FAILURE

CORRELATION FOR SUPPORTING DATA:
  1) EVENT CODE ACS78893
  2) EVENT CODE 00801204
  3) OPERATION PRIORITY 3

ENTER A (ACTION)

???
CMD==>

```

Figure 208. Host NetView - Event Details (Page 1 of 2)

If you press the *Enter* key, you can see the second of two pages for Event Details as shown in Figure 209 on page 217.

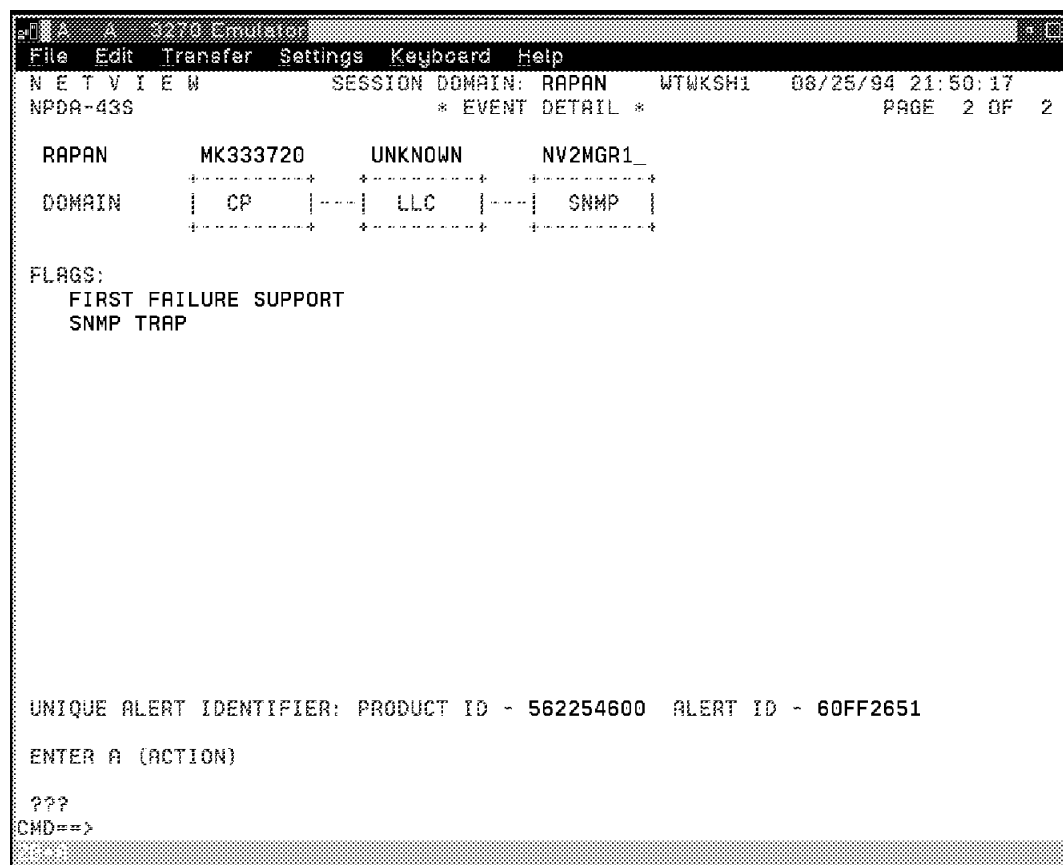


Figure 209. Host NetView - Event Details (Page 2 of 2)

If you return to the *Event Detail Menu* by pressing *PF3*, and choose option 2 for *Product Set Identification Display*, you will see a screen identifying what management software is being used at the management station to forward alerts to host NetView. This is shown in Figure 210 on page 218.

```

A A 3270 Emulator
File Edit Transfer Settings Keyboard Help
N E T V I E W          SESSION DOMAIN: RAPAN      WTWKSH1    08/25/94 21:48:22
NPD8-44B                * SENDER SOFTWARE PRODUCT ID *          PAGE 1 OF 4

RAPAN      MK333720      UNKNOWN      NV2MGR1_
+-----+ +-----+ +-----+
| CP      |---| LLC      |---| SNMP      |
+-----+ +-----+ +-----+

DATE/TIME: 08/25 21:18

PRODUCT CLASSIFICATION          IBM PROGRAMMING

SERVICEABLE COMPONENT IDENTIFIER  562254600 (PRODUCT ID)
COMPONENT RELEASE LEVEL           200
SOFTWARE COMMON NAME              IBM NetView for OS/2

???
CMD==>

```

Figure 210. Host NetView - Product Set Identification

9.1.8 Initiating Actions from NetView for MVS Using RUNCMD

In addition to being able to receive and monitor all of your LAN-based alerts on a host NetView console, you can also initiate OS/2 commands from any NetView command line.

9.1.8.1 Prerequisites for Using the NetView RUNCMD Command

In order for you to send commands and receive output back to your NetView console, you must have the following components of Communications Manager/2 installed and running on the target OS/2 machine:

1. Service Point Application Router (SPAR)
2. Remote Operations Service (ROPS)

When you choose to selectively install ROPS during a Communications Manager/2 installation, you will get both SPAR and ROPS. One icon for each will be placed in your Communications Manager/2 main folder. You should start SPAR first. It will open a window when it is started. After that you should start ROPS. It will also create a window for itself and then a line item showing the *REMOTEOP* application will appear in the SPAR window. You can now minimize both windows.

9.1.8.2 Using RUNCMD to Issue OS/2 and LMU Commands

You are now ready to start sending commands to a selected OS/2 station using RUNCMD. Following is the syntax for the RUNCMD command:

RUNCMD (NCCF)
Syntax

```
>> RUNCMD SP=spname,NETID=net_id,APPL=applname,OP=oper_id; command
```

Figure 211. Host NetView RUNCMD Syntax

- *spname* - Is the service point name. This is the local node name that we set up in Figure 181 on page 195. We had set up *MK333720* in that window, so that is what we use here.
- *net_id* - Is the Network ID name. This is the local network name that we also set up in Figure 181 on page 195. We set this to *USIBMMK* in that window, so that is what we used here.
- *applname* - In host NetView terms, this specifies the name of the link connection subsystem manager (LCSM) used to process the command. In CM/2 terms, this is the application name running under SPAR. In the previous section, we started ROPS to run under SPAR. Therefore, the application name is simply *REMOTEOP*.
- *oper_id* - Is the operator ID name. You can enter your own NetView operator ID. We just left it blank and it defaulted to our own user ID.
- *command* - This can be any OS/2 command. The maximum length of a command is 240 characters.

Now that we know all of the syntax, we are ready to enter our sample RUNCMD command. In order to test the communications, we started with a simple command *TYPE C:\CONFIG.SYS*. The full RUNCMD that we entered is shown at the bottom of Figure 212 on page 220.

```

NCCF      N E T V I E W      RAPAN WTWKSH1 08/25/94 22:03:58
* RAPAN   RUNCMD SP=MK333720,NETID=USIBMMK,APPL=REMOTEOP,OP=;NET START REQ
-         Start of Output [MK333720] NET START REQ
-
-         The REQUESTER service is starting.....
-         The REQUESTER service was started successfully.
-         (C) Copyright IBM Corporation 1988, 1992. All rights reserved.
-         (C) Copyright Microsoft Corporation 1988, 1991. All rights
-         reserved.
-
-         End of Output [MK333720] NET START REQ
* RAPAN   RUNCMD SP=MK333720,NETID=USIBMMK,APPL=REMOTEOP,OP=;NET WHO
-         Start of Output [MK333720] NET WHO
-
-         Users on Domain NVSRVDM
-
-         User ID          Requester          Time since      Comment
-         -----
-         DANNO            NVSRV30            00:41:46      Test
-         LMUMGR           WTR33372        00:00:56      LMU Managing
-         The command completed successfully.
-
-         End of Output [MK333720] NET WHO
-----
-         /T:D:\ANV2\ETC\DBASE\BTRIEVE.TRN
-         SET NMSADMIN=IPXSEARCH
-         DEVICE=D:\THESEUS2\THESEUS2.SYS
-
-         End of Output [MK333720] TYPE C:\CONFIG.SYS
-
-         ???
runcmd sp=mk333720,netid=usibmmk,appl=remoteop,op=;type c:\config.sys

```

Figure 212. Host NetView - Using RUNCMD to Enter OS/2 and LAN Commands

Notice that the output scrolls on the screen. In Figure 212, we started OS/2 LAN Requester, and checked to see who was logged on.

```
runcmd sp=mk333720,netid=usibmmk,appl=remoteop,op=;NET START REQ
```

```
runcmd sp=mk333720,netid=usibmmk,appl=remoteop,op=;NET WHO
```

Since we are sending commands to a machine that is also an LMU managing station, we can use *LMUCMD* to issue commands to remote machines that have the LMU agent code.

As shown in Figure 213 on page 221, the LMU managing machine is displayed as being logged on as *SUPERVISOR*. An OS/2 station next to it on the ring is identified by its LAN computername of *WTR32216*.

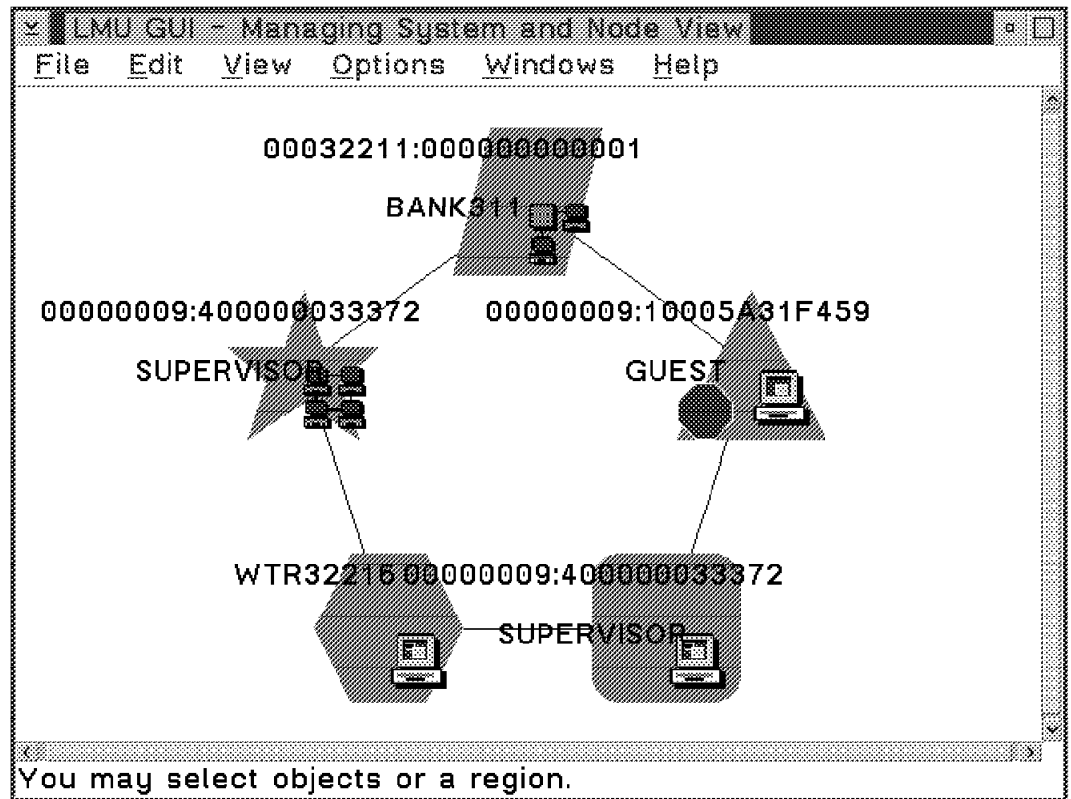


Figure 213. LMU Network of Managed Machines

Now that we know the LMU address of the remote OS/2 station, we can issue the LMU QUERYVPD command to obtain a listing of the installed vital product data. The RUNCMD and its output are shown in Figure 214 on page 222.

```

A - A - 3270 Emulator
File Edit Transfer Settings Keyboard Help
NCCF N E T V I E W RAPAN WTKSH1 08/25/94 22:24:35
**
-
-   Operating System ..... OS/2 2.11
-
-   *****
-   *'
-   |                               Hardware Configuration
-   |
-   |*****
-   **
-
-   Machine Type ..... IBM PS/2 Model 80
-   Product Number ..... 8580-071
-   Processor ..... Intel 80386
-   Processor Speed ..... 16 MHz
-   CoProcessor ..... Intel 80387
-   Bus Type ..... Micro Channel 32-Bit
-
-   Total Memory ..... 14336 KB = 14.0 MB
-
-   Equipment List ..... 1 Parallel Port(s)
-   1 Serial Port(s)
-   1 Diskette Drive(s)
-   2 Fixed Disk(s)
-   Math CoProcessor
-
-   Serial Port 1 ..... COM1
-   Baud rate ..... 1200 bps
-   Data bits ..... 7
-
-   ??? ***
-   runcmd sp=mk333720,netid=usibmmk,appl=remoteop,op=;lmucmd wtr32216 queryvdp
-

```

Figure 214. Using LMU LMUCMD to Issue Commands to Remote Stations

9.1.9 Automating Actions from NetView for MVS

In this section, we will show how to set up some automation in a host NetView environment. For a complete description of how to implement automated actions using host NetView, please refer to *NetView Automation Implementation*, LY43-0015. We will briefly describe the concepts of MSUs, alert vectors, automation tables and automation routines. In order to make it easier to understand, we will describe each of these topics through the use of a scenario.

Our scenario is quite simple:

- We will be monitoring a workstation to ensure that the OS/2 LAN Requester is up and running.
- When the Requester service goes down, an alert is sent to the NetView for OS/2 managing station.
- The alert is parsed by the TRALERT daemon. It will see that it is a LAN Server/Requester alert (as shown in 9.1.5.1, “Modifying the Supplied Alert Filter File” on page 206) and then send it to host NetView for further processing.
- When it gets to host NetView, it is parsed in the automation table and the major vector will be analyzed.
- The automation table determines that it is an alert from our particular station, and that it is a LAN Requester alert specifying that the Requester service has gone down.
- The automated action for this particular alert will be to change the display characteristics of this alert on the NetView console to blinking and RED and call a CLIST to do the following:
 - Issue a command back to our workstation to start the Requester service.
 - Present a pop-up message to the workstation saying that host NetView has restarted the Requester on its behalf.

9.1.9.1 MSUs, Alert Vectors and Automation

There are many different types of data that are used in automation. The most common are messages and commands. Starting with Version 2.2 of host NetView, you could also process data in the form of a management services unit. This *management services unit*, or MSU, is a data structure that NetView uses to manage system or network resources. The type of MSU that interests us is the network management vector transport (NMVT). The NMVT is an envelope for transporting *major vectors*, which actually contain the management services data. An example of a management services major vector is the *alert* major vector, which is identified with a key of X'0000'.

NOTE

In order to work with alerts being processed by host NetView, you should be fairly proficient in reading and deciphering hexadecimal notation. You will be viewing the alert in hex, and based on that content, you can code an automation routine to resolve the situation.

Before coding an automation table entry, you will have to decide which messages and MSUs to automate. Typically, you would check with your NetView operators to review what goes on during a typical shift and then identify which

MSUs lead to a predictable sequence of commands. In our case, we know that we want to trap on the situation where LAN Requester goes down so we can bring it up again.

9.1.9.2 Extracting MSU Information from NetView Screens

We will now start our scenario showing the alert coming into the host NetView console. We will then use the alert to build an entry in the automation table, so that if the alert comes in again, we can trap on it and initiate an automation routine to resolve our problem.

As soon as the LAN Requester goes down the alert is received by the NetView for OS/2 managing station. Since the Host Connection program is running with our filter to pass LAN Server/Requester alerts, it will pass the alert to the host NetView console as shown in Figure 215.

```

B 3270 Emulator
File Edit Transfer Settings Keyboard Help
NETVIEW SESSION DOMAIN: RAPAN WTWKSH1 08/31/94 15:23:45
NPDQ-308 * ALERTS-STATIC *

SEL# DOMAIN RESNAME TYPE TIME ALERT DESCRIPTION:PROBABLE CAUSE
( 1) RAPAN RS60003 DEV 15:23 SNMP RESOURCE PROBLEM: UNDETERMINED
( 2) RAPAN RS60003 DEV 15:22 SNMP RESOURCE PROBLEM: UNDETERMINED
( 3) RAPAN RS60003 DEV 15:22 SNMP RESOURCE PROBLEM: UNDETERMINED
( 4) RAPAN RS60003 DEV 15:22 SNMP RESOURCE PROBLEM: UNDETERMINED
( 5) RAPAN NV2MGR1 SNMP 15:22 SNMP RESOURCE PROBLEM: UNDETERMINED
( 6) RAPAN RS60003 DEV 15:22 SNMP RESOURCE PROBLEM: UNDETERMINED
( 7) RAPAN RS60003 DEV 15:21 SNMP RESOURCE PROBLEM: UNDETERMINED
( 8) RAPAN RS60003 DEV 15:21 SNMP RESOURCE PROBLEM: UNDETERMINED
( 9) RAPAN RS60003 DEV 15:21 SNMP RESOURCE PROBLEM: UNDETERMINED
(10) RAPAN RS60003 DEV 15:21 SNMP RESOURCE PROBLEM: UNDETERMINED
(11) RAPAN NV2MGR1 SNMP 15:21 SNMP RESOURCE PROBLEM: UNDETERMINED
(12) RAPAN RS60003 DEV 15:20 SNMP RESOURCE PROBLEM: UNDETERMINED
(13) RAPAN RS60003 DEV 15:20 SNMP RESOURCE PROBLEM: UNDETERMINED
(14) RAPAN RS60003 DEV 15:19 SNMP RESOURCE PROBLEM: UNDETERMINED
(15) RAPAN RS60003 DEV 15:19 SNMP RESOURCE PROBLEM: UNDETERMINED
(16) RAPAN RS60003 DEV 15:18 SNMP RESOURCE PROBLEM: UNDETERMINED
(17) RAPAN RS60003 DEV 15:18 SNMP RESOURCE PROBLEM: UNDETERMINED
(18) RAPAN ITS082 DEV 15:18 PROBLEM RESOLVED: REMOTE NODE
(19) RAPAN ITS082 DEV 15:18 PROBLEM RESOLVED: COMMUNICATIONS INTERFACE
(20) RAPAN RS60003 DEV 15:18 SNMP RESOURCE PROBLEM: UNDETERMINED
(21) RAPAN RS60003 DEV 15:17 SNMP RESOURCE PROBLEM: UNDETERMINED
(22) RAPAN RA7NCKH COMC 15:16 CONFIG/CUSTOMIZATION ERR: CONFIGURATION
(23) RAPAN RA7NCKH COMC 15:16 CONFIG/CUSTOMIZATION ERR: CONFIGURATION
DEPRESS ENTER KEY TO VIEW ALERTS-DYNAMIC OR ENTER A TO VIEW ALERTS-HISTORY
ENTER SEL# (ACTION), OR SEL# PLUS M (MOST RECENT), P (PROBLEM), DEL (DELETE)

???
CMD=> 11

```

Figure 215. LAN Requester Alert on Host NetView Console

Notice our alert is on line 11 on the Alerts-Static screen. To get more information about our alert, we type in **11** and press the *Enter* key. As shown in the previous section, this will present us with the *Recommended Action for Selected Event* screen as shown in Figure 216 on page 225.

```

IBM - 3270 Emulator
File Edit Transfer Settings Keyboard Help
N E T V I E W      SESSION DOMAIN: RAPAN      WTWKSH1      08/31/94 15:27:32
NPD9-45A          * RECOMMENDED ACTION FOR SELECTED EVENT *      PAGE 1 OF 1
RAPAN            MK333720      UNKNOWN      NV2MGR1_
DOMAIN          +-----+ +-----+ +-----+
                | CP      | |---| LLC      | |---| SNMP      |
                +-----+ +-----+ +-----+

ACTIONS - I008 - PERFORM PROBLEM DETERMINATION PROCEDURE FOR ORIGINATOR
          9.24.104.54
          I588 - IF REQUIRED, QUERY CONTACT ID Mirek Iwachow AT LOCATION
                NAME Raleigh bld 062 room L610 ABOUT COMPONENT ID nv2mgr1.i
                tso.ral.ibm.com

ENTER ST (MOST RECENT STATISTICS), DM (DETAIL MENU), OR D (EVENT DETAIL)
???
CMD==> DM

```

Figure 216. Host NetView Recommended Action for LAN Requester Alert

Figure 216 gave us the first piece of information that we will need for building our automation table entry. This screen tells us *who* the alert came from. Therefore, when we get a LAN Requester alert from 9.24.104.54, we will need to start the recovery routine.

However, we would like even more detail than this, so we type in *DM* and press the *Enter* key. This will give us the Detailed Menu as shown in Figure 217 on page 226.

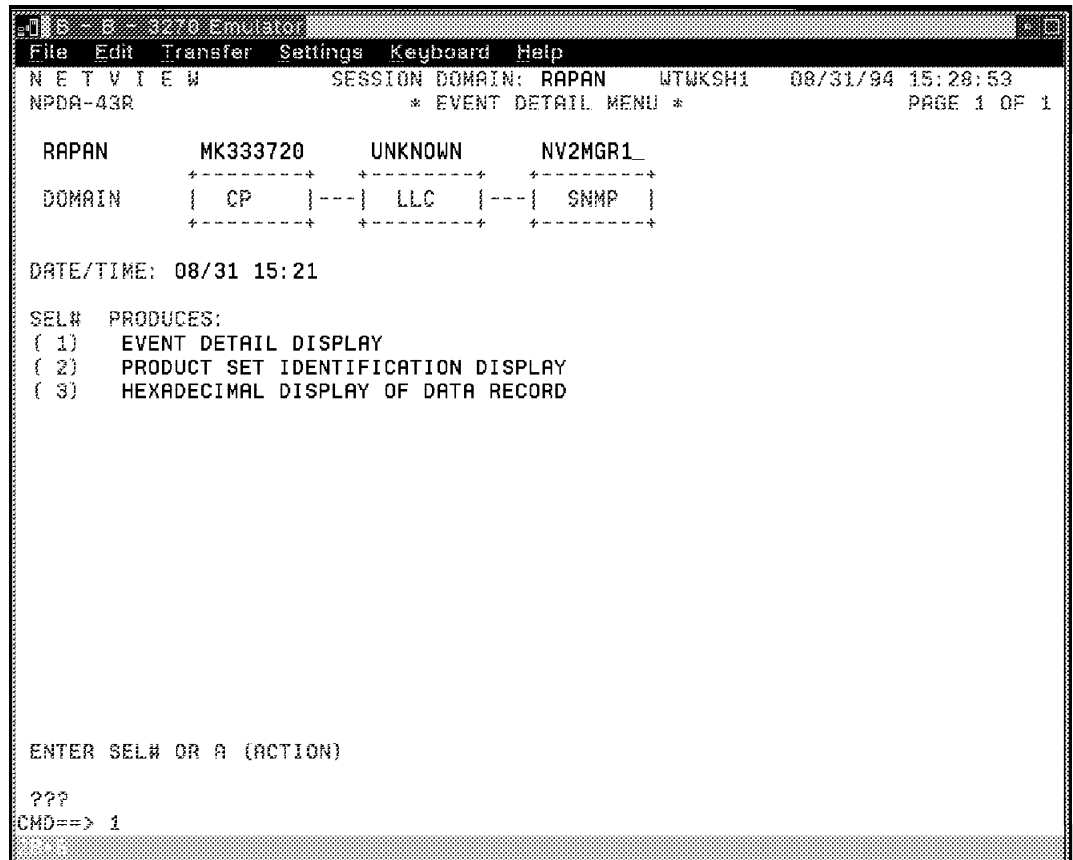


Figure 217. Host NetView - Event Detail Menu for LAN Requester Alert

From the Event Detail Menu, we are interested in two things:

1. The Event Detail Display
2. The Hexadecimal Display of our Data Record

First, we will look at the Event Detail Display as shown in Figure 218 on page 227.

```

C B 3270 Emulator
File Edit Transfer Settings Keyboard Help
N E T V I E W          SESSION DOMAIN: RAPAN   WTWKSH1   08/31/94 15:20:29
NPDA-43S                * EVENT DETAIL *                PAGE 1 OF 2

RAPAN      MK333720      UNKNOWN      NV2MGR1_
+-----+ +-----+ +-----+ +-----+
DOMAIN     | CP |---| LLC |---| SNMP |
+-----+ +-----+ +-----+ +-----+

DATE/TIME: RECORDED - 08/31 15:21   CREATED - 08/31/94 15:20:02

EVENT TYPE: UNKNOWN

DESCRIPTION: SNMP RESOURCE PROBLEM

PROBABLE CAUSES:
    UNDETERMINED

QUALIFIERS:
    1) ENTERPRISE 1.3.6.1.4.1.2.6.58
    2) SNMP GENERIC-TRAP NUMBER ENTERPRISE SPECIFIC
    3) SNMP SPECIFIC-TRAP NUMBER 1
    4) SNMP MIB VARIABLE VALUE 0

CORRELATION FOR SUPPORTING DATA:
    1) EVENT CODE ACS66727
    2) EVENT CODE 00801204

ENTER A (ACTION)

???
CMD==>

```

Figure 218. Event Details for LAN Requester Alert

We can now start to recognize that the alert is in fact a LAN Requester alert. Refer to the section of the screen that shows the four alert *QUALIFIERS*. We will address each one and show how it will help qualify the alert that we want to trap on.

1. Enterprise 1.3.6.1.4.1.2.6.58

Note that this is the dot notation Class identifier for LAN Requester.

2. SNMP Generic-Trap Number *ENTERPRISE SPECIFIC*

3. SNMP Specific-Trap Number 1

4. SNMP MIB Variable Value 0

Note that this MIB Variable is set to a 0. This means that the LAN Requester has gone down. If this value is non-zero (for example, set to a 3), then it is up and running fine.

We now have enough information to positively identify this as a LAN Requester problem alert and we can use this information to build our automation table entry. An outline of what we would like to trap on is shown in Figure 219 on page 228.

```

IF ((performing problem determination procedures
    for an IP address is desired) AND
    (the IP Address is 9.24.104.54)) THEN
BEGIN
  IF (trap contains a SysObjectID (CLASS ID)          AND
      SysObjectID (CLASS ID) is 1.3.6.1.4.1.2.6.58      AND
      trap contains a SNMP Generic Trap Binding        AND
      SNMP Generic Trap Binding is 'ENTERPRISE SPECIFIC' AND
      trap contains a SNMP Specific Trap Number        AND
      SNMP Specific Trap Number is 1                  AND
      trap contains a SNMP MIB Variable Value          AND
      SNMP MIB Variable Value is 0 ))
  THEN
    * Highlight the Alert Message on the Console in RED
    * Execute a Command List to Restart Requester and
      Display a Message POPUP
END

```

Figure 219. Pseudo Code of our Automation Table Entry

9.1.9.3 Coding the Automation Table Entry

Now that we know *what* we want to do, we have to translate our pseudo-code into something that NetView can understand in its automation table. As mentioned earlier, the alert comes to NetView as a major vector. This vector can be broken down into subvectors, which in turn, can be broken down further into subfields.

For example, each of the four *QUALIFIERS* described above come from the *Detailed Data* subvector. Each subvector has an associated hex representation. The Detailed Data subvector is known as an X'98' subvector. The subfields of an X'98' subvector are known as network alert common subfields and represented in hex as X'82' in the subfield.

Remember that an alert is a major vector which is identified with a key of X'0000'. If you wanted to reference a segment of this MSU alert data record in subfield X'82' of subvector X'98', you would use the *MSUSEG* syntax as shown in the following example:

```
IF (MSUSEG(0000.98.82) = '0' THEN ...action
```

Before coding the automation table entry, it would be useful to take a look at a couple of tables that describe the two subvectors that we will need to take segments of using the *MSUSEG* operator. The two subvectors are:

1. Cause Undetermined (X'97') subvector
2. Detailed Data (X'98') subvector

Table 3. Cause Undetermined (X'97') Subvector Code Points and Text	
Code Point	Text
X'00B0'	Perform Problem Determination Procedures For (detailed data qualifier).
X'31D0'	If Required, Query sysContact at Location Name sysLocation About Component ID sysName

Table 4. Detailed Data (X'98') Subvector Data ID, Data Types and Text		
Data ID	Data Type	Comments
X'F8'	Enterprise	The sysObjectID or CLASS ID binding
X'FA'	SNMP Generic Trap Number	The generic trap binding number for SNMP traps 0 through 6. Hex representation for COLD START, WARM START, LINK DOWN, LINK UP, AUTHENTICATION FAILURE, EGP NEIGHBOR LOSS, or ENTERPRISE SPECIFIC.
X'FB'	SNMP Specific Trap Number	The specific trap binding. The enterprise specific trap number.
X'FC'	SNMP MIB Variable Name	The MIB variable name coming from the standard MIB (I or II) or from an enterprise specific extension to the MIB.
X'FD'	SNMP MIB Variable Value	The SNMP MIB Variable Value is sent when something has be defined in the preceding subfield (MIB Variable Name)

We are now ready to start coding our automation table entry. By looking at the pseudo-code in Figure 219 on page 228 and at the two tables above, we can see how each MSUSEG operation will point to the exact subvector and subfield so that we can do our condition checking. Following is the required coding that needs to be inserted into the automation table to trap on the LAN Requester alert from our workstation. This code is shown in Figure 220.

```

IF ((MSUSEG(0000.97.81(1)) = . HEX('00B0') . ) &
    (MSUSEG(0000.97.82(1) 6) = '9.24.104.54' . )) THEN
  BEGIN ;
    IF ((MSUSEG(0000.98.82(1) 4) = HEX(' F8') . ) &
        (MSUSEG(0000.98.82(1) 6) = '1.3.6.1.4.1.2.6.58' . ) &
        (MSUSEG(0000.98.82(2) 4) = . HEX(' FA') . ) &
        (MSUSEG(0000.98.82(2) 6) =
          . HEX(' C5D5E3C5D9D7D9C9E2C500E2D7C5C3C9C6C9C3') . ) &
        (MSUSEG(0000.98.82(3) 4) = HEX(' FB') . ) &
        (MSUSEG(0000.98.82(3) 6) = '1' . ) &
        (MSUSEG(0000.98.82(4) 4) = HEX(' FD') . ) &
        (MSUSEG(0000.98.82(4) 6) = '0' . ))
      THEN
        XHILITE(BLI) COLOR(RED)
        EXEC(CMD(' STRTREQ') ROUTE(ONE WTWKSH2)) ;
    END ;

```

Figure 220. Section of Host NetView AUTOTBL for LAN Alerts

NOTE

When we were checking the SNMP generic trap (HEX 'FA'), we were testing to see if MSUSEG (0000.98.82(2) 6) was equal to 'ENTERPRISE SPECIFIC', but found an X'00' instead of X'40' (which is a blank) separating these 2 words. Therefore, we had to specifically check against the HEX representation of 'ENTERPRISE SPECIFIC' which is: HEX('C5D5E3C5D9D7D9C9E2C500E2D7C5C3C9C6C9C3') as shown above.

We can now explain how the MSUSEG operator works with parameters. Remember that we had *four* Qualifiers describing our LAN Requester trap. Each on these qualifiers creates a new X'82' subfield. Within each subfield, you may

have an offset that will take you to the start of the detailed data. In our example, the SNMP specific trap number is in the third occurrence of the X'82' subfield and is offset to byte 6. Therefore, the MSUSEG conditional check will be:

IF (MSUSEG(0000.98.82(3) 6) = '1' .)

The period (.) after the '1' means to ignore all subsequent data.

You will want to check your MSUSEG coding against the actual subvectors. To do this, go back to your Event Detail Menu and choose Selection 3 and press the *Enter* key as shown in Figure 221.

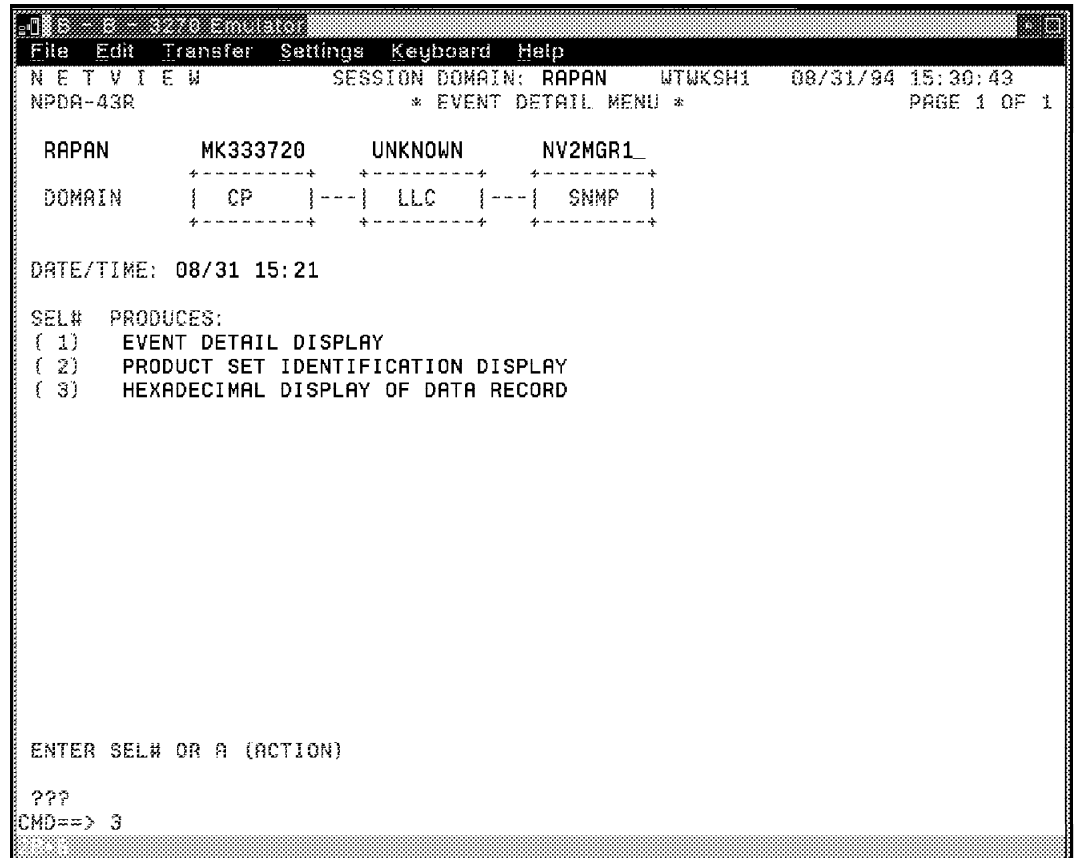


Figure 221. Selecting to See Hexadecimal Display of Subvectors

This will take you to the *Hexadecimal Display of Data Record* screen as shown in Figure 222 on page 231.

```

B - B - 3270 Emulator
File Edit Transfer Settings Keyboard Help
N E T V I E W          SESSION DOMAIN: RAPAN   WTWKSH1   08/31/94 15:31:34
NPDA-44C               * HEXADECIMAL DISPLAY OF DATA RECORD *          PAGE 1 OF 2

RAPAN      MK333720      UNKNOWN      NV2MGR1_
+-----+ +-----+ +-----+ +-----+
DOMAIN     | CP      |---| LLC      |---| SNMP      |
+-----+ +-----+ +-----+ +-----+

DATE/TIME: 08/31 15:21

MSU:
MAJOR VECTOR 0000 - 018E 0000
SUBVECTOR 01
0A010810 5E081F0F 1402
SUBVECTOR 05
36053410 0009D4D2 F3F3F3F7 F2F000F4 09E4D5D2 D5D6E6D5 40403B19 95A5F294
8799F14B 89A3A296 4B998193 4B898294 4B839694 00FC
SUBVECTOR 10
65100027 11040E02 F5F6F2F2 F5F4F6F0 F0F2F0F0 1606C9C2 D400D585 A3E58985
A6008696 9900D6E2 61F20D11 03130011 F9F5F9F5 F1D5E30B 110E0804 F0F0F0F0
F0F02311 03200ED5 85A3E589 85A60086 969900D6 E261F200 8281A285 00E2D5D4
D7008187 85
SUBVECTOR 92
0B921800 12B00CDD B3159A
SUBVECTOR 97
6A970681 00B031D0 10820039 11F94BF2 F44BF1F0 F44BF5F4 128200F9 11D48999
859200C9 A6818388 96A62082 004B11D9 81938589 87880000 82938400 F0F6F200
00999696 9400D3F6 F1F01D82 00331195 A5F29487 99F14B89 A3A2964B 9981934B
8982944B 83969403 8300

???
CMD==>

```

Figure 222. Hexadecimal Display of Data Record (Page 1 of 2)

This is the first of two pages. This page shows subvectors 01, 05, 10, 92 and 97. Note that subvector X'97' does in fact have the X'00B0' code point starting in byte 5 and that the IP address starts in byte 13 (X'F94BF2F4...).

If you press the *Enter* key, you can see the second of two pages that shows subvector X'98' as shown in Figure 223 on page 232.

```

B 3270 Emulator
File Edit Transfer Settings Keyboard Help
N E T V I E W      SESSION DOMAIN: RAPAN      WTWKSH1  08/31/94 15:32:12
NPDA-44C          * HEXADECIMAL DISPLAY OF DATA RECORD *      PAGE 2 OF 2

  RAPAN      MK333720      UNKNOWN      NV2MGR1_
  +-----+ +-----+ +-----+ +-----+
  | CP | |---| | LLC | |---| | SNMP |
  +-----+ +-----+ +-----+ +-----+

DATE/TIME: 08/31 15:21

SUBVECTOR 98
3D981782 61F811F1 4BF34BF6 4BF14BF4 4BF14BF2 4BF64BF5 F8188261 FA11C5D5
E3C5D9D7 D9C9E2C5 00E2D7C5 C3C9C6C9 C3068261 FB11F106 8261FD11 F0
SUBVECTOR 93
0493FE00
SUBVECTOR 96
0D960401 10000481 30E10383 61
SUBVECTOR 48
22480D82 000911C1 C3E2F6F6 F7F2F70D 82000911 F0F0F8F0 F1F2F0F4 068200BC
11F3

???
CMD==>

```

Figure 223. Hexadecimal Display of Data Record (Page 2 of 2)

The Class ID for LAN Requester is in the first subfield of subvector X'98'. Notice that the hex display of 1.3.6.1.4.1.2.6.58 starts in byte 8 (X'F14BF34BF6...). The other subfields follow that and are led by their respective data IDs (X'FA', X'FB' and X'FD')

9.1.9.4 Starting and Testing Our Automation Table Entry

Now that we have designed our automation table entry, we would like to test the syntax and start the automation table with our entry. We placed our table entry in a separate data set which was to be included in the main automation table data set. For example, if your automation table was in a data set called 'NETVIEW.V02R04M0.DSIPARM(ATBPROD)' and your NetView for OS/2 table entries are in a member called NV4OS2, you can then imbed NV4OS2 into ATBPROD by adding a line to the ATBPROD member:

```
%INCLUDE NV4OS2
```

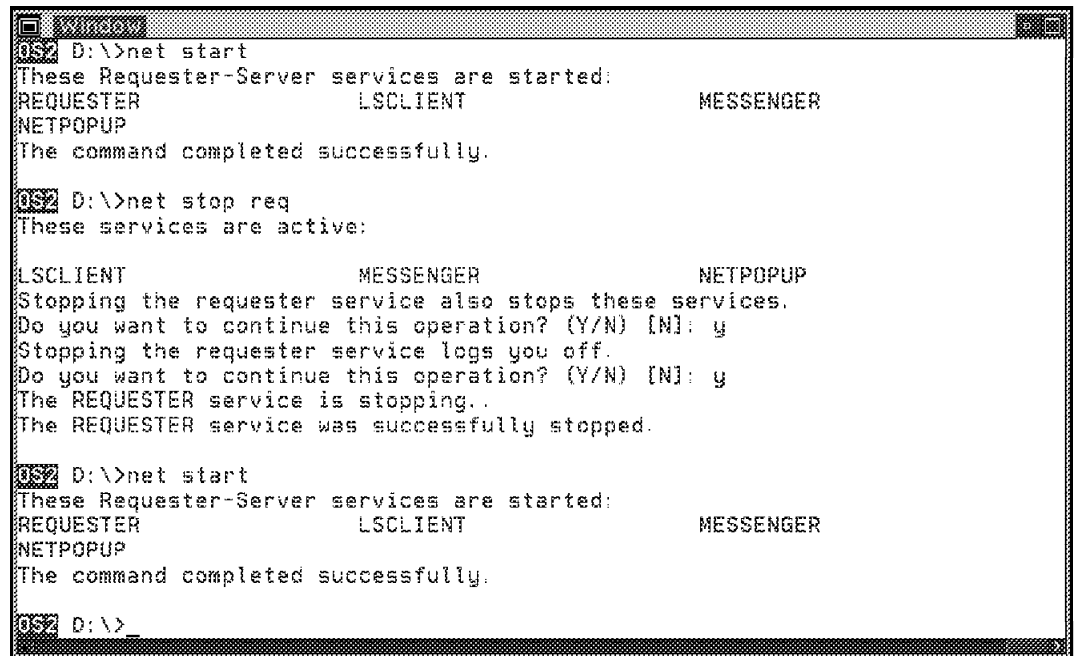
To test the syntax of the entry that we coded in Figure 220 on page 229, issue the following AUTOTBL command from a NetView console:

```
AUTOTBL MEMBER=ATBPROD,TEST
```

If this is successful, you can start the automation table with your new entries using the following AUTOTBL command:

```
AUTOTBL MEMBER=ATBPROD
```

Now that the automation table has been started, we are ready to test the entire scenario. If we go to our LAN Requester workstation and start up an OS/2 prompt window, we can check to see if the Requester is up and running and then issue the net stop req command as shown in Figure 224.



```
OS/2 D:\>net start
These Requester-Server services are started:
REQUESTER          LSCLIENT          MESSENGER
NETPOPUP
The command completed successfully.

OS/2 D:\>net stop req
These services are active:

LSCLIENT          MESSENGER          NETPOPUP
Stopping the requester service also stops these services.
Do you want to continue this operation? (Y/N) [N]: y
Stopping the requester service logs you off.
Do you want to continue this operation? (Y/N) [N]: y
The REQUESTER service is stopping..
The REQUESTER service was successfully stopped.

OS/2 D:\>net start
These Requester-Server services are started:
REQUESTER          LSCLIENT          MESSENGER
NETPOPUP
The command completed successfully.

OS/2 D:\>
```

Figure 224. Stopping OS/2 LAN Requester and Host NetView Restarting It

At this point, an alert will be generated and sent to the NetView for OS/2 managing station, where it will be forwarded up to host NetView. We can see that our automation table entry has worked, because the line item for our LAN Requester alert (line 12) is highlighted as shown in Figure 225 on page 234.

```

A - A - 3270 Emulator
File Edit Transfer Settings Keyboard Help
NETVIEW SESSION DOMAIN: RAPAN WTWKSH1 08/31/94 17:24:10
NPDA-300 * ALERTS-STATIC *

SEL# DOMAIN RESNAME TYPE TIME ALERT DESCRIPTION:PROBABLE CAUSE
( 1) RAPAN RA6003CP*DEV 17:24 SNMP RESOURCE PROBLEM:UNDETERMINED
( 2) RAPAN RA6003CP*DEV 17:23 SNMP RESOURCE PROBLEM:UNDETERMINED
( 3) RAPAN RA6003CP*DEV 17:23 SNMP RESOURCE PROBLEM:UNDETERMINED
( 4) RAPAN RA6003CP*DEV 17:23 SNMP RESOURCE PROBLEM:UNDETERMINED
( 5) RAPAN RA6003CP*DEV 17:23 SNMP RESOURCE PROBLEM:UNDETERMINED
( 6) RAPAN RA6003CP*DEV 17:22 SNMP RESOURCE PROBLEM:UNDETERMINED
( 7) RAPAN NV2MGR1_ SNMP 17:21 SNMP RESOURCE PROBLEM:UNDETERMINED
( 8) RAPAN RA6003CP*DEV 17:21 SNMP RESOURCE PROBLEM:UNDETERMINED
( 9) RAPAN RA6003CP*DEV 17:21 SNMP RESOURCE PROBLEM:UNDETERMINED
(10) RAPAN RA6003CP*DEV 17:21 SNMP RESOURCE PROBLEM:UNDETERMINED
(11) RAPAN RA6003CP*DEV 17:20 SNMP RESOURCE PROBLEM:UNDETERMINED
(12) RAPAN NV2MGR1_ SNMP 17:20 SNMP RESOURCE PROBLEM:UNDETERMINED
(13) RAPAN RA6003CP*DEV 17:19 SNMP RESOURCE PROBLEM:UNDETERMINED
(14) RAPAN RA6003CP*DEV 17:19 SNMP RESOURCE PROBLEM:UNDETERMINED
(15) RAPAN RA6003CP*DEV 17:19 SNMP RESOURCE PROBLEM:UNDETERMINED
(16) RAPAN RA6003CP*DEV 17:19 PROBLEM RESOLVED:REMOTE NODE
(17) RAPAN RA6003CP*DEV 17:19 PROBLEM RESOLVED:COMMUNICATIONS INTERFACE
(18) RAPAN RA6003CP*DEV 17:19 SNMP RESOURCE PROBLEM:UNDETERMINED
(19) RAPAN RA6003CP*DEV 17:19 SNMP RESOURCE PROBLEM:UNDETERMINED
(20) RAPAN RA6003CP*DEV 17:18 SNMP RESOURCE PROBLEM:UNDETERMINED
(21) RAPAN RA6003CP*DEV 17:18 SNMP RESOURCE PROBLEM:UNDETERMINED
(22) RAPAN RA6003CP*DEV 17:18 SNMP RESOURCE PROBLEM:UNDETERMINED
(23) RAPAN RA7NCKH COMC 17:18 CONFIG/CUSTOMIZATION ERR:CONFIGURATION

DEPRESS ENTER KEY TO VIEW ALERTS-DYNAMIC OR ENTER A TO VIEW ALERTS-HISTORY
ENTER SEL# (ACTION),OR SEL# PLUS N (MOST RECENT), P (PROBLEM), DEL (DELETE)

???
```

Figure 225. Automation Table Entry Highlights LAN Requester Alert

The second thing we added in our table entry was to execute a CLIST. The line in our table entry was:

```
EXEC(CMD('STRTREQ') ROUTE(ONE WTWKSH2))
```

This will execute the STRTREQ CLIST. Since we are not always logged on we would like this CLIST to be executed by an automation operator. The operator ID that we used for automation was WTWKSH2. The *route* command will route our STRTREQ CLIST to this operator and execute it. The content of our CLIST is shown in Figure 226.

```

SYSTEM1.CLISTS(STRTREQ) - 01.05 Columns 00001 00072
***** Top of Data *****

RUNCMD SP=MK333720,NETID=USIBMMK,APPL=REMOTEOP,OP=;NET START REQ

RUNCMD SP=MK333720,NETID=USIBMMK,APPL=REMOTEOP,OP=;LMUPOPUP /R /Ii +
"NetView Service" "NetView for MVS has Re-Started the Requester"

***** Bottom of Data *****
```

Figure 226. Command List Called From our Automation Table Entry

The CLIST uses the facilities of the RUNCMD to execute OS/2, LAN and LMU commands at our workstation. These commands restart the LAN Requester service and display a pop-up at the Requester workstation.

The STRTREQ CLIST executed on the WTWKSH2 NetView console is shown in Figure 227.

```

NCCF N E T V I E W RAPAN WTWKSH2 08/31/94 17:22:18
C RAPAN RUNCMD SP=MK333720,NETID=USIBMMK,APPL=REMOTEOP,OP=;NET START REQ
- Start of Output [MK333720] NET START REQ
-
- The REQUESTER service is starting.....
- The REQUESTER service was started successfully.
- (C) Copyright IBM Corporation 1988, 1992. All rights reserved.
- (C) Copyright Microsoft Corporation 1988, 1991. All rights
- reserved.
-
- End of Output [MK333720] NET START REQ
C RAPAN RUNCMD SP=MK333720,NETID=USIBMMK,APPL=REMOTEOP,OP=;LMUPOPUP /R /Ii
- "NetView Service" "NetView for MVS has Re-Started the Requester"
- [MK333720] Command ID 0 is long-running.
C RAPAN DSI013I COMMAND LIST STRTREQ COMPLETE

```

???

Figure 227. STRTREQ CLIST Running on WTWKSH2 NetView Operator Console

As soon as the Requester service was restarted, an lmupopup message was sent to the Requester workstation as shown in Figure 228.

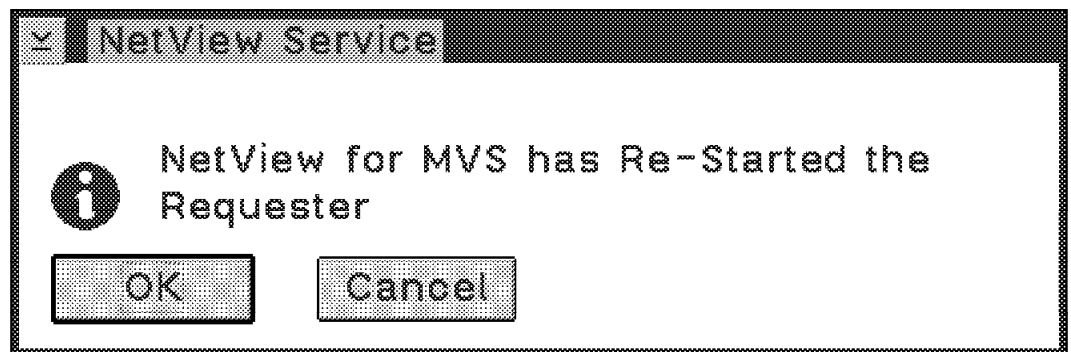


Figure 228. LMUPOPUP Message Stating Requester Has Been Restarted

Finally, if we go back to our OS/2 Command Prompt window as shown in Figure 224 on page 233, and issue **NET START**, we will see that the Requester service has been successfully restarted by NetView for MVS.

Chapter 10. Exploiting REXX Programs and the SNMP Commands

The Data Collector, Event Automator and Application Builder are powerful tools, but there may be times when you are required to do things that these tools can not. You can create your own REXX applications to automate network management tasks, retrieve complex information from MIB variables or store values in a DB2/2 database. By incorporating SNMP commands into your REXX programs you can customize network administration to suit your needs.

This chapter shows some user examples of how you can use REXX to:

1. Store values retrieved from an SNMP MIB into a DB2/2 database.
2. Collect combinations of numerics, strings and table values and display them on the screen.
3. Create complex threshold and rearm conditions.
4. Automate changes to MIB variables for a groups of systems.
5. Interface with LMU.

10.1 Storing MIB Values in a Database

NetView for OS/2 does not require DB2/2; however, LMU/2 does use the DB2/2 database to store data from its managed systems. Once data is stored in a DB2/2 database it can be retrieved in an organized manner, used by other programs, or used to create reports. This can help in network management and planning.

The following REXX program shows how to create your own database table and populate it using a seed file. Error checking has been reduced to a minimum to keep the program simple. Knowledge of SQL is required to be able to create commands which are used to create and manipulate the database. You can modify the program to store data in the LMU/2 database or any other database. The SNMP commands (SNMPGET, SNMPSET, SNMPWALK, SNMPNEXT and SNMPTRAP) used to retrieve data from the host system using SNMP are located in the \anv2\bin\ directory. The SQL and Database Manager commands and libraries used in REXX are part of the DB2/2 product.

The example used a seed file for the systems that were to be stored in the database. The name of the seed file is entered at the command line when running the REXX program. The seed file is a plain ASCII file in the same format as the seed file or log file for TCP/IP discovery. To run the REXX program create the seed file first using any text editor. You can use host names or the dotted decimal notation. Anything after the # will be ignored. The example below used the following seed file:

```
nv2mgr1          # 9.24.104.54
nv2mgr2          # 9.24.104.55
nv2client        # 9.24.104.68
nv2anet4         # 9.24.104.105
```

Following is a listing of the program.

```

/*REXX*/
/*****
/* register DATABASE MANAGER functions      */
/* register SQL functions                    */
*****/
return_code = rxfuncadd('sqldbs','sqlar','sqldbs')
return_code = rxfuncadd('sqlexec','sqlar','sqlexec')
/*****
/* START DATABASE MANAGER                  */
/* return errors                          */
/* error code number                      */
*****/
call sqldbs 'start database manager'
      if (result<>0) then say result
      if (sqlca.sqlcode<>0) then say 'start dbm' sqlca.sqlcode
/*****
/* Create SNMPS database on D drive        */
/* return errors                          */
/* error code number                      */
*****/
call sqldbs 'create database snmps on d'
      if (result<>0) then say result
      if (sqlca.sqlcode<>0) then say 'Create db' sqlca.sqlcode
/*****
/* LOG to SNMPS database in Shared MODE   */
/* return errors                          */
/* error code number                      */
*****/
call sqldbs 'start using database snmps in shared mode'
      if (result<>0) then say result
      if (sqlca.sqlcode<>0) then say 'start db' sqlca.sqlcode
/*****Creating a Table*****/
/* SQL command for creating a new table... */
/* table_name= TESTTBL and column names    */
/* are: ID, HOST, LOCATION, and SYSUPTIME  */
/* Primary key is ID                      */
/* Concatenate cmlx sections into cml      */
/* Call sql command ":" in front of cml is */
/* intentional                            */
*****/
cmla = 'create table testtbl '
cmlb = '(id smallint not null,'
cm1c = 'host char(25) not null,'
cm1d = 'location char(40) not null,'
cm1f = 'primary key(id))'
cm1=cmla||cm1b||cm1c||cm1d||cm1f
call sqlexec 'execute immediate :cm1'
/*****
/* return errors                          */
/* error code number                      */
*****/
      if (result<>0) then say result
      if (sqlca.sqlcode<>0) then do
          say 'create table' sqlca.sqlcode
          say 'create table' sqlca.sqlstate
      end
@echo off
/*****

```

```

/* Read in seed file and mib objects to put      */
/* in data base                                  */
/* format :                                       */
/* SNMPREXX.CMD filename.log object1            */
/*****/
arg file obj1
/* count is unique id variable                  */
count = 1
/*****/
/* while file is not empty                      */
/* read in line and separate name and ip address */
/* execute snmpget for the variables            */
/* queue result into RXQUEUE                   */
/* pull result off of queue                    */
/* separate info from value. store value in     */
/* instanceval                                  */
/*****/
do while lines(file) > 0
    parse value linein(file) with name "#" ipaddr
    'call snmpcmd snmpget' name obj1 '| rxqueue'
    pull sysname
    parse var sysname mibinstance ":" instancetype ":" instanceval
/*****/
/* create insert command                        */
/* concatenate command parts                   */
/*****/
    cm2a = 'insert into testtbl '
    cm2b = "values("||count|| ","|| name||","||instanceval ||")"
    cm2=cm2a||cm2b
/*****/
/* execute sql statement and report errors      */
/*****/
    call sqlexec 'execute immediate :cm2'
    if (result<>0) then say result
    if (sqlca.sqlcode<>0) then do
        say 'insert row ' sqlca.sqlcode
        say 'insert row state' sqlca.sqlstate
    end

count = count + 1
end
/*****/
/* Housekeeping : close database and manager   */
/*****/
call sqldbs 'stop using database'
call sqldbs 'stop database manager'

```

To run our program we typed: `SNMPREXX d:\anv2\bin\seedfile.txt`. The program created a table called TESTTBL in a DB2/2 database called SNMP. You can of course change these values to suit your needs.

We then ran the Query Manager to display the results. Figure 229 on page 240 shows the results of the query.

ID	HOST	LOCATION
1	nv2mgr1	ITSC RALEIGH, BLUE DRIVE 1024, OFFICE
2	nv2mgr2	ITSC RALEIGH, BLUE DRIVE 1024, OFFICE
3	nvclient	ITSC RALEIGH, BLUE DRIVE 1024, OFFICE
4	nv2anet4	RALEIGH BLD HALL

*** END ***

Figure 229. Query of TESTTBL Table in Query Manager

10.2 Collecting and Displaying Combinations of MIB Variable Types

The MIB Browser and Data Collector can display and store values for any MIB variable, however, combinations of MIB types are not allowed. For example, displaying a table, string or value must be done separately. You can write a REXX program that displays all these value on the screen all at the same time. The following example displays the values stored in any three MIB variables given. It can be any combination of table, string, and numeric. We entered the following to display the disk storage used (table), system location (string) and system up time (numeric ticks):

```
SNMPREX2 nv2mgr1 system.host..... system.location system.sysuptime
```

The following is the program that displays the values of the above MIB variables to the screen.

```

/*REXX*/
'@echo off'
/*****
/* Program queries a given host system for MIB variables */
/* can be any combination of table, number or string */
**
** Command syntax:
**
**      ┌──────────┐ ┌──────────┐ ┌──────────┐ ┌──────────┐ ┌──────────┐
**      │ drive    │ │ path     │ │          │ │          │ │          │
**      └──────────┘ └──────────┘ └──────────┘ └──────────┘ └──────────┘
**
**      ┌──────────┐ ┌──────────┐ ┌──────────┐ ┌──────────┐ ┌──────────┐
**      │ name     │ │ obj1    │ │          │ │          │ │          │
**      └──────────┘ └──────────┘ └──────────┘ └──────────┘ └──────────┘
**
**      Where:
**      name      IP name of system to be polled
**      obj1      MIB instance (table, number or string)
**      obj2      MIB instance (table, number or string)
**      obj3      MIB instance (table, number or string)
**
*****/
/* parse arguments in command line */
/*****
arg name obj1 obj2 obj3
*****/
/* push variables retrieved from MIB onto queue */
/*****
if obj1<>' then 'call snmpcmd snmpwalk' name obj1 '| rxqueue'
if obj2<>' then 'call snmpcmd snmpwalk' name obj2 '| rxqueue'
if obj3<>' then 'call snmpcmd snmpwalk' name obj3 '| rxqueue'
say 'Name of system:' name
*****/
/* pull values off of queue here you can store */
/* the values individually from the queue and store */
/* them in an array called mibvalues */
/*****
count=1
do while queued() > 0
    pull line
    parse var line mibinstance ":" instancetype ":" mibvalues.count
    count = count +1
end
*****/
/* Print out the array value in sequential order */
/*****
do loopcounter = 1 to count-1
    say loopcounter ' = ' mibvalues.loopcounter
end

```

The results of running this program are shown in Figure 230 on page 242.

```

D:\anv2\bin>snmprex2 nv2mgr1 host.hrstorage.hrstorageetable.hrstorageentry.hr
storagedescr system.syslocation system.sysuptime
Name of system: NV2MGR1
1 = RAM
2 = C:FIXED DISK
3 = D:FIXED DISK
4 = E:FIXED DISK
5 = F:FIXED DISK
6 = T:FIXED DISK (REMOTE)
7 = D:VIRTUAL MEMORY/SWAPPED DISK
8 = ITSC RALEIGH, BLUE DRIVE 1024, OFFICE H1001A
9 = (1211500) 3:21:55.00
D:\anv2\bin>_

```

Figure 230. Multiple Types of MIB Values

10.3 Creating Complex Thresholds and Sending Traps

The Data Collector can poll, collect a MIB value and send traps to the managing system based on thresholds and rearm values that have been set. The limitation is that only one threshold or rearm value for a single MIB instance can be set. To create more complex thresholds and traps, you have to write your own programs. By writing your own REXX programs for polling, thresholds and traps, you can:

- Poll combinations of MIB variables and set thresholds when these combinations exceed limits. For example, when the sum of free space on all fixed disks of a system exceeds a certain value take some action.
- Using *IF...THEN...* statements, you can send traps when a combination of MIB values has exceeded user specified thresholds. For example, CPU utilization on a 386 system would be different than on a 486 system, so traps would depend on the type of system and its CPU utilization. This could be determined in the *IF...THEN...* statements before a trap is sent.

The following program dynamically retrieves all storage devices on a system, isolates the fixed disks on the system and then polls only the SNMP data (to reduce SNMP traffic) for these fixed disks at a given interval. It then adds up the disk's capacity and their used disk space. Then it creates a ratio and compares it with a given threshold. It will display a pop-up and send a trap when the threshold is exceeded.

```

/*REXX*/
/* Starts a process that polls a given system */
/* displays a pop-up a window when total disk */
/* fixed disk space on the system exceeds a */
/* specified percentage
**
** Command syntax:

```



```

        count = count + 1
    end
    /*****
    /* loop while not Cancel key pressed
    /* retrieve used disk space for all entries in the fixed array
    /* and add them up to create used_sum.
    *****/
    key_pressed = 1
    do while key_pressed <> 2
        count = 1
        used_sum = 0
        do while count < num
            temp = '.1.3.6.1.2.1.25.2.3.1.6.'||fixed.count
            'call snmpcmd snmpget' name temp 'rxqueue'
            pull line
            parse var line mibinstance ":" instancetype ":" size
            used_sum = used_sum + size
            count = count + 1
        end
        /*****
        /* compare percentage full with specified value in threshold
        /* display a popup if exceeded
        *****/
        if ((used_sum/size_sum*100) > threshold) then do
            message1 = 'Disk space below 20% on '||name||'! Now at '
            message2 = used_sum||' KB out of '||size_sum||' KB'
            message = message1||message2
            key_pressed = rxmessagebox(message, "Warning!", "OKCANCEL", "Warning")
        end
        /* wait polling interval and start over */
        call sysleep poll
    end

```

When the threshold is exceeded a trap will be sent to a managing system which can be configured as a specific trap and a pop-up will appear on the system running this program. When the threshold was exceeded the message box in Figure 231 was displayed.

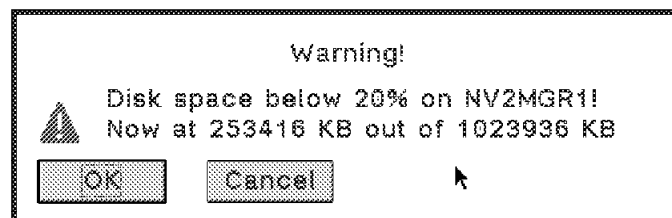


Figure 231. Pop-Up Message for Exceeded Disk Capacity

10.4 Automating Changes to a Group of Systems

There are times when changes must be made to MIB values. The MIB Browser can be used for this, however, when similar changes must be made to a whole group of systems, it would be inefficient to use the MIB Browser. An example of this would be when a department is relocated to another building or location. The MIB instance syslocation has to be changed to reflect the new address and office number for each person that has moved. You can create a REXX program

to automate these changes. The program can retrieve the old values, change only certain sections, and save the changes. The list of people and their new locations can be retrieved from another database or a text file created by a spreadsheet. The *syslocation* values before running this program were similar to the ones shown in Figure 232.

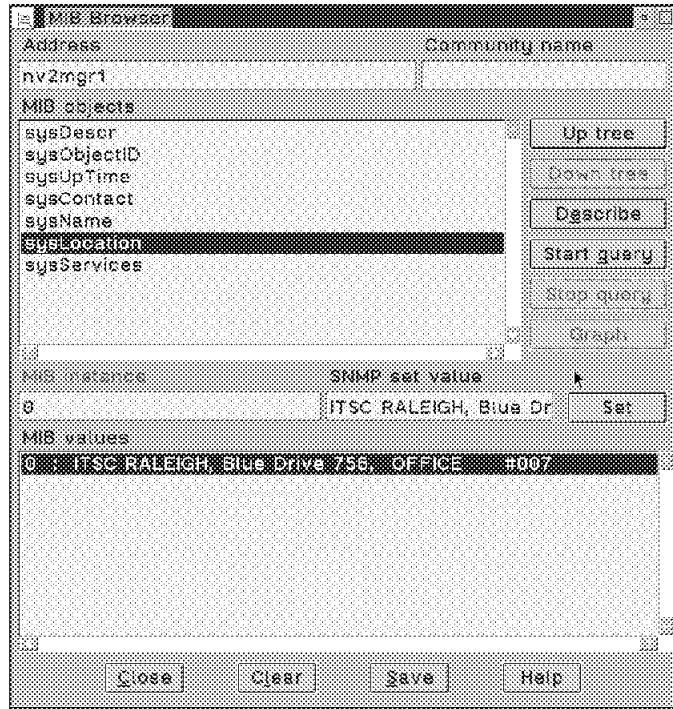


Figure 232. Value of Syslocation before Changes

The text file used in this program to change the syslocation for one group of systems is shown below:

```
Green Drive 345
nv2mgr1, 123B
nv2mgr2, 345C
nvclient, 7077
```

The following example changes the **syslocation** of systems listed in a text file according to a specified pattern.

```

/*REXX*/
ECHO OFF
/*****
/* program which uses a seed file with new address of a department at*/
/* the top and host_name,new_office number for each moved employee */
*****/
**
** Command syntax:
**
**      ┌─── drive ───┐ ┌─── path ───┐ ──── SNMPREX4 ────>
**      └──────────┘ └──────────┘
**
**      ─────────── file ───────────┘
**
** Where :
**
**      file      ascii file containing data as follows
**
**              newaddress
**              host, new_number
**              host, new_number
**              host, new_number
**              .      .
**              .      .
**              .      .
*****/
/* get file name into REXX program */
/*****

ARG file
/*****
/* read in new address , dept. moved to another street or location */
/*****
new_address = linein(file)
/*****
/* While the file is not empty */
/* read in a new host name and the new office number from file */
/* put in the que this host's old information */
/* pull data from que */
/* split up old data into its components: department,address, */
/* room description and office number */
/* create new location data by combining the old and the new */
/* store it back in the host */
/* end loop */
/*****
DO WHILE lines(file) > 0
  PARSE VALUE linein(file) WITH name "," new_number
  'CALL SNMPCMD SNMPGET' name 'system.syslocation.0 | RXQUEUE'
  PULL sysname
  PARSE VAR sysname mibinstance ':' instancetype ':' department
  "," address "," room_description "#" office_number
  new_location = "'"||department||'", '||new_address||',
  '||room_description||' #'||new_number||'"
  'CALL SNMPCMD SNMPSET' name 'system.syslocation.0 octetstring'
  new_location
END

```

After the program runs, the MIB Browser is used to view the *syslocation* variable to check that the changes were made. See Figure 233 on page 247.

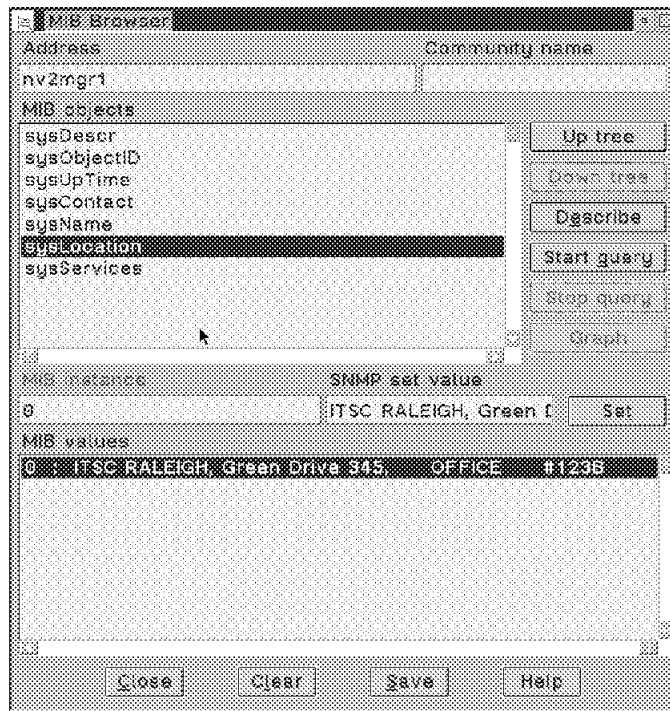


Figure 233. SYSLOCATION after the REXX Program Has Executed

Chapter 11. Scenario 1 - Managing a Lexmark 4039 SNMP Printer

In this scenario we will show how to integrate an enterprise-specific MIB into your NetView for OS/2 management environment, and then monitor that device for traps. We have chosen a Lexmark 4039-16L laser printer which we installed at the ITSO in Raleigh. This 4039 printer comes equipped with its own MIB. This MIB is in a file called *lexmark1.mib*. We will explain how you can acquire this MIB in a later section.

In this scenario, we will show the following:

- Load the MIB into our NetView for OS/2 managing system.
- Use the MIB Browser to find what MIB variables are of interest.
- Set up the Data Collector to monitor printer status.
- Generate an alert when the status changes.
- Show the alert on the Event Displayer.
- Integrate it into the Event Automator to forward the alert to NetView for MVS for further action.

11.1 Loading the Lexmark MIB into Our NetView for OS/2 Managing System

We acquired the MIB (*lexmark1.mib*) in much the same manner as we obtained the Cisco MIB as explained in 3.2.1, "Obtaining the CISCO Router MIB via FTP" on page 60. This time we used the facilities of TCP/IP FTP to access Lexmark's machine in Lexington, Kentucky. Here are the steps you can take to get the *lexmark1.MIB*:

1. Use the TCP/IP FTP program to access the machine where the MIB was located. This was `FTP.LEXMARK.COM`.
2. Log in as an Anonymous user with any password.
3. Issue the FTP get command to copy the MIB to the required OS/2 directory:
`GET /pub/mib/lexmark1.mib d:\anv2\snmp_mib\lexmark1.mib`
4. Quit out of the FTP session to return to an OS/2 command prompt.

Once you have the MIB file in your `\anv2\snmp_mib` directory, you can issue the following command to load the MIB into your existing NetView for OS/2 standard MIB tree:

```
LOADMIB -load lexmark1.mib
```

Once the MIB is loaded, choose the MIB Browser from the NetView for OS/2 folder to start looking at the MIB for interesting variables to monitor. If your MIB loading exercise was successful, you can start traversing down the tree starting at the *private* level then go down to **enterprises**. You should eventually arrive at a MIB Variable window showing at least the following two enterprises:

```
ibm
lexmark
```

We now had three enterprise-specific MIBs loaded into our system as shown in Figure 234 on page 250.

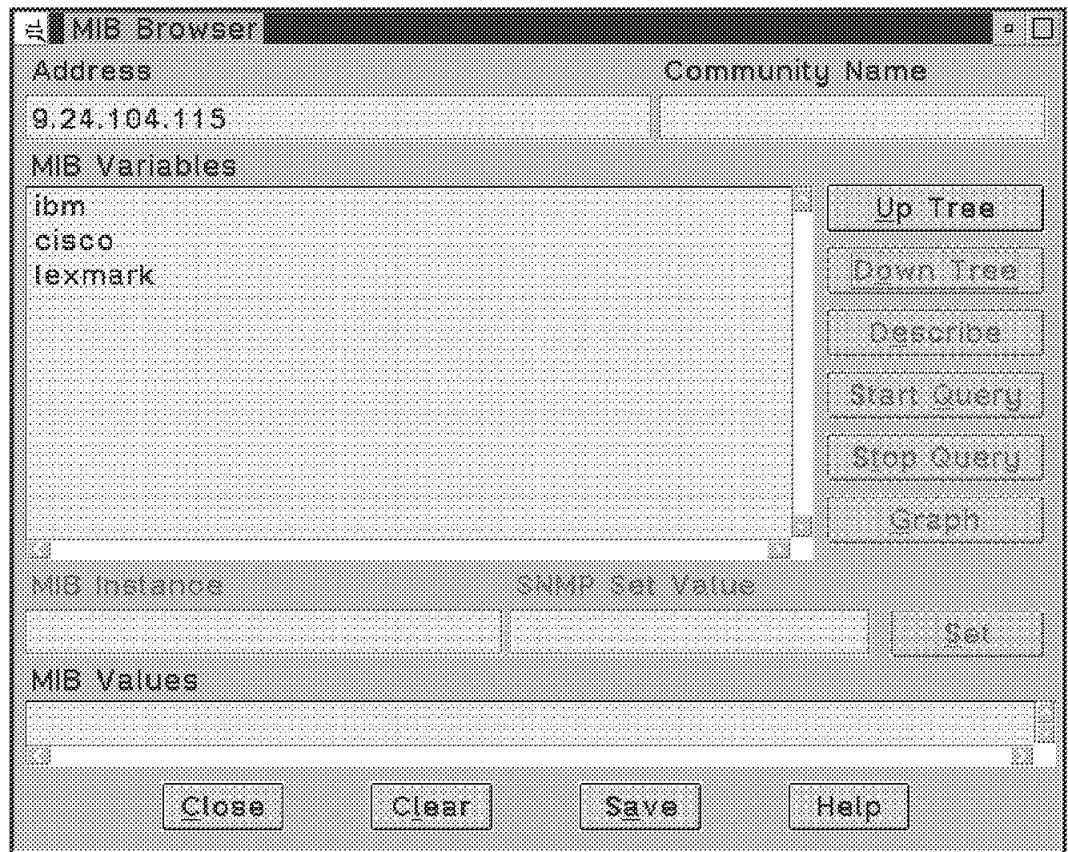


Figure 234. MIB Browser Showing Enterprise-Specific MIBs

11.2 Using the MIB Browser to Find Interesting Variables

The first thing we wanted to do was explore the MIB, looking for those variables that we would like to monitor. These are the variables that can provide you with meaningful status information that you can monitor. If something is not running as desired, then you can trap this information and cause some automated action to correct the situation. In the case of our Lexmark 4039-16L printer, we went down the following path in the MIB tree to find *prtgenStatusEntry*:

```
* private
  enterprises
    lexmark
      printer
        prtgen
          prtgenStatusTable
            prtgenStatusEntry
```

From here, we went down one more level to find some interesting status information such as *prtgenStatusBusy* as shown in Figure 235. Note that when you query this line item by clicking on the *Start Query* button, you get the following value displayed in the *MIB Values* window:

1 : Not-Busy

If the printer was very busy when you started your query, you would have got the following response:

2 : Busy

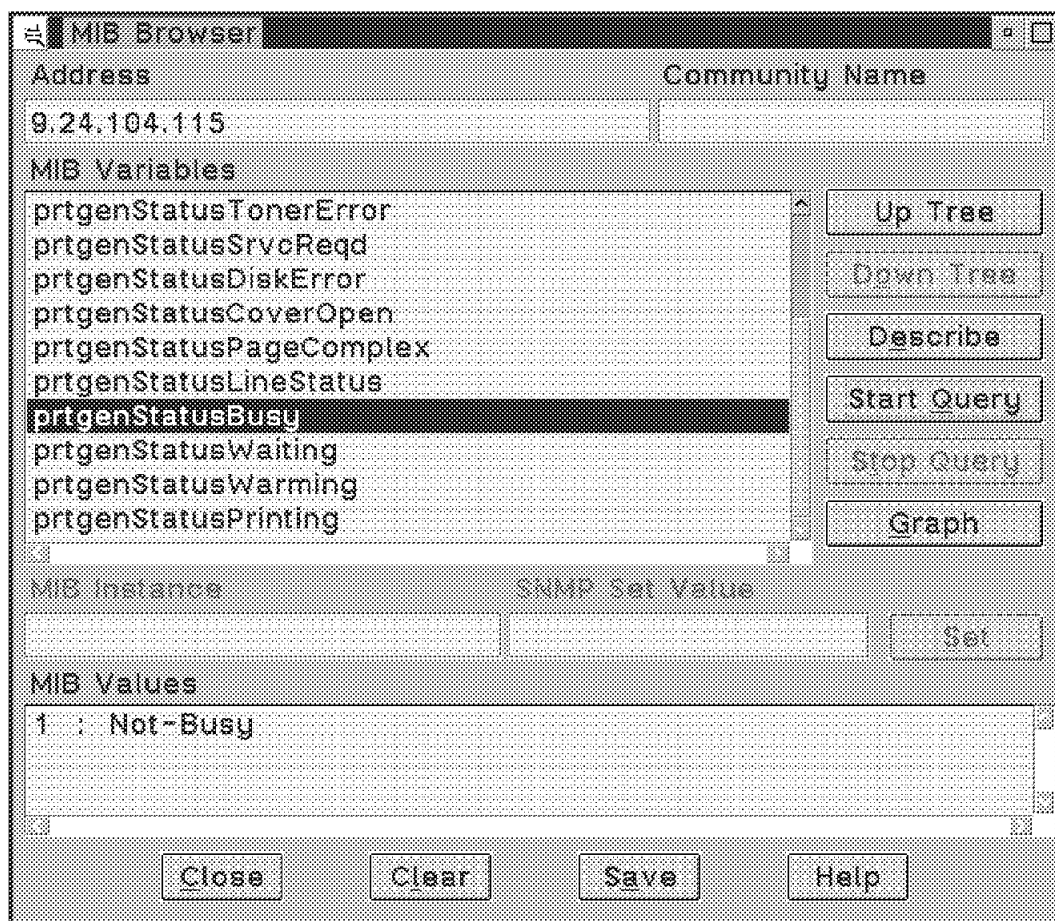


Figure 235. 4039 Status MIB Variables

Important Note

For all of the Lexmark Status variables, you will get a 1 for a Normal condition, and a 2 for an Abnormal condition. This is important because you can set a threshold on any of the status items. For example, if you set a threshold value of 1 on the *prtgenStatusBusy* item, an alert would be generated as soon as the status changed to a 2 indicating the printer is busy.

We will show how to generate a similar type of alert on another status variable in 11.4, “Generating an Alert When the Status Changes” on page 257.

Following is a list of Status Indicators and their associated values that are available on the Lexmark 4039-16L printer:

<i>Table 5. IBM 4039-16L LaserPrinter MIB Status Variables</i>	
MIB Variable - Status Indicator	Associated Values
prtgenStatPrinterIndex	1 (only one printer)
prtgenStatusIRC	0 (current intervention required)
prtgenStatusOutHopFull	1 : Hopper-Not-Full
	2 : Hopper-Full
prtgenStatusInputEmpty	1 : Input-Not-Empty
	2 : Input-Empty
prtgenStatusPaperJam	1 : No-Jam
	2 : Jam
prtgenStatusTonerError	1 : No-Toner-Error
	2 : Toner-Error
prtgenStatusSrcvReqd	1 : No-Service-Required
	2 : Service-Required
prtgenStatusDiskError	1 : No-Disk-Error
	2 : Disk-Error
prtgenStatusCoverOpen	1 : Cover-Closed
	2 : Cover-Open
prtgenStatusPageComplex	1 : Page-OK
	2 : Page-Too-Complex
prtgenStatusLineStatus	1 : Online
	2 : Offline
prtgenStatusBusy	1 : Not-Busy
	2 : Busy
prtgenStatusWaiting	1 : Not-Waiting
	2 : Waiting
prtgenStatusWarming	1 : Not-Warming
	2 : Warming
prtgenStatusPrinting	1 : Not-Printing
	2 : Printing

In this scenario, we are going to monitor the prtgenStatusInputEmpty MIB variable. This variable tells us whether or not we are out of paper. When everything is normal and there is plenty of paper in the paper tray, we can query this MIB variable and get the results as shown in Figure 236 on page 253.

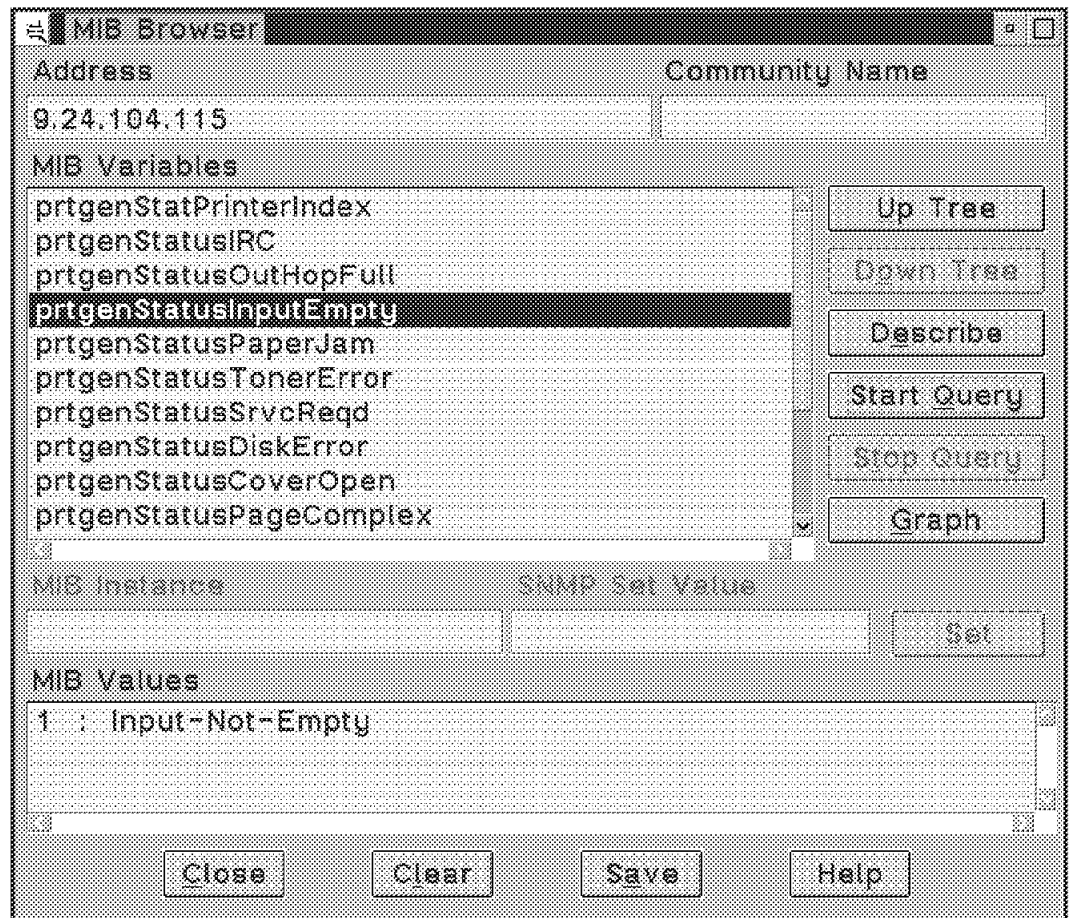


Figure 236. Status of InputEmpty MIB Variable

Note that the MIB value is set to '1' meaning that the input tray is not empty.

11.3 Setting Up the Data Collector to Monitor Printer Status

As mentioned in a previous section, we would like to set a threshold on the `prtgenStatusInputEmpty` variable such that when it changes from a 1 to a 2 an alert is generated and sent to the managing station. Then it will be sent to another SNMP manager.

We are going to use the facilities of the Data Collector to:

- Monitor the value of this variable.
- Show the data collected.
- Show this data in a graph format.
- Set the threshold, and generate an alert.

We start by bringing up the MIB Data Collector main window by clicking on the **Data Collector** icon on the NetView for OS/2 folder. On this main window, we click on the *Add...* button, which will give us the *Add MIB Object Selection* window. Both of these windows are shown in Figure 237 on page 254. In the *Add MIB Object Selection* window, we are presented with the MIB tree which we can traverse just as we did in the previous section to find the `prtgenStatusInputEmpty` variable as shown in Figure 237 on page 254.

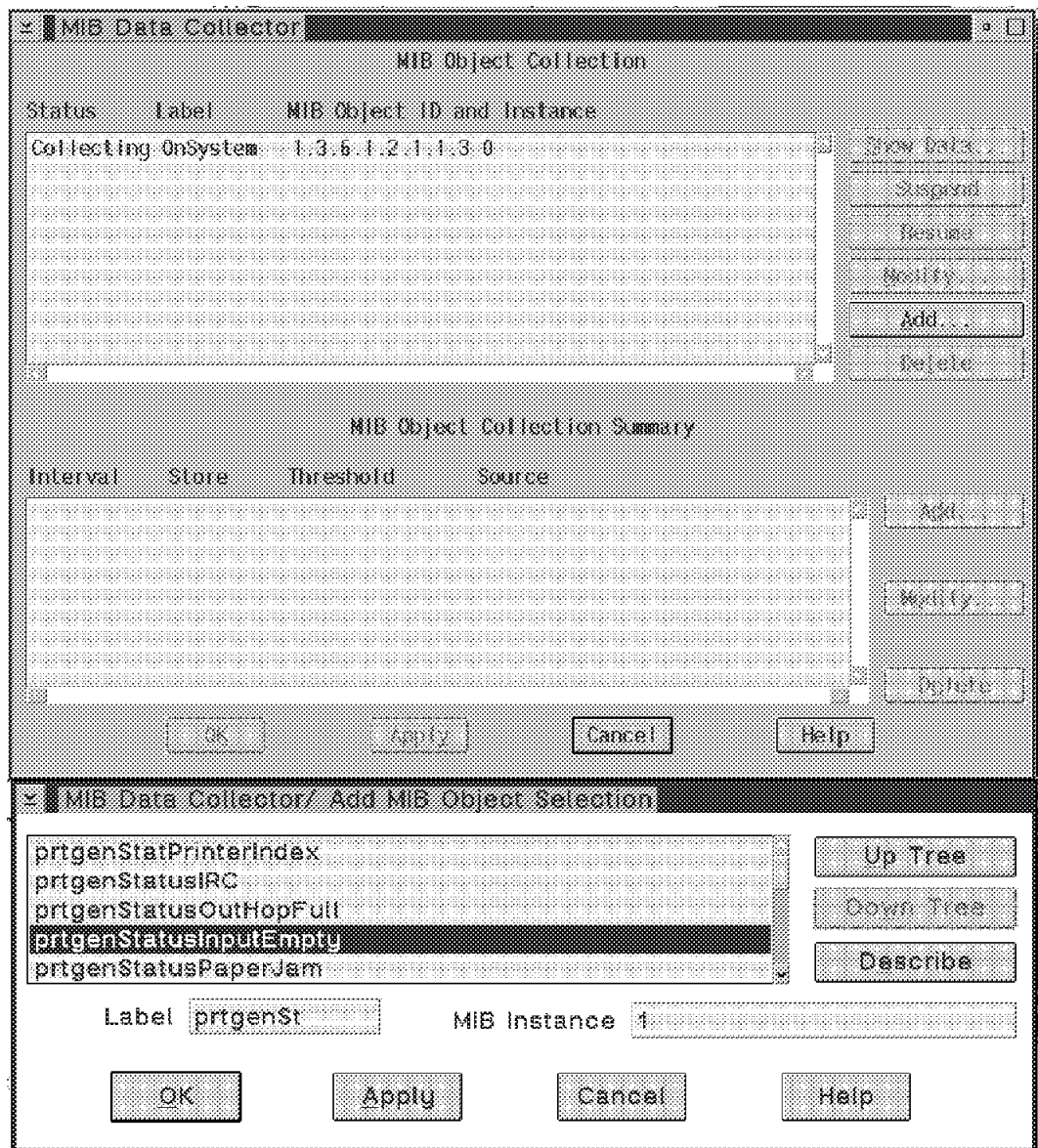


Figure 237. Adding the InputEmpty MIB Variable to Data Collector

When we chose this variable, it prompted us to enter or change the *Label* box contents and enter the MIB instance. The *Label* name defaults to the first 8 characters of the MIB variable name. We did not change it. Since there is only one instance of *prtgenStatusInputEmpty* (there is only one printer at this time), we typed in the number *1* in the *MIB Instance* field. The Data Collector will create a file called `\anv2\collect\prtgenSt.1` to store the data in once we begin collecting. As you can see, this file name is the concatenation of the *Label* and *MIB Instance* fields.

Now that we're all set, we press the *OK* button, and the application will present us with the *Add Summaries for prtgenSt* window where we can set the threshold information. We enter the address of our 4039 printer which is 9.24.104.115 in the Host Name or Address field. We took all the defaults except for the Threshold and Rearm fields. We typed in a 1 in both fields because the threshold is strictly greater than 1, and we are trapping on a value of 2. The rearm value is less than or equal to 1, which is what we get when the situation

goes back to normal. One final setting is to change the rearm specification to be an absolute number as opposed to a percentage of the threshold number. This is done by clicking on the **Absolute** radio button. All the above entries are shown in Figure 238.

The screenshot shows a window titled "MIB Data Collector/ Add Summaries for prtgenS1". It contains several input fields and controls:

- Host Name or Address:** A text field containing "9.24.104.115".
- Collection Mode:** A dropdown menu showing "Store, Check Thresholds".
- Polling Interval:** A text field containing "30s".
- Trap Number:** A text field containing "58720263".
- Threshold >:** A text field containing "1".
- Rearm <=:** A text field containing "1".
- Radio Buttons:** Two radio buttons labeled "Percent" and "Absolute". The "Absolute" button is selected.
- Threshold Action:** A text field.
- Rearm Action:** A text field.
- Buttons:** Three buttons at the bottom: "OK", "Cancel", and "Help".

Figure 238. Setting the Polling Interval and Thresholds on InputEmpty

Now that we have entered everything into this window, we can click on the *OK* button and we are brought back to the MIB Data Collector main window showing our line item and that it is collecting data. This is shown in Figure 239 on page 256.

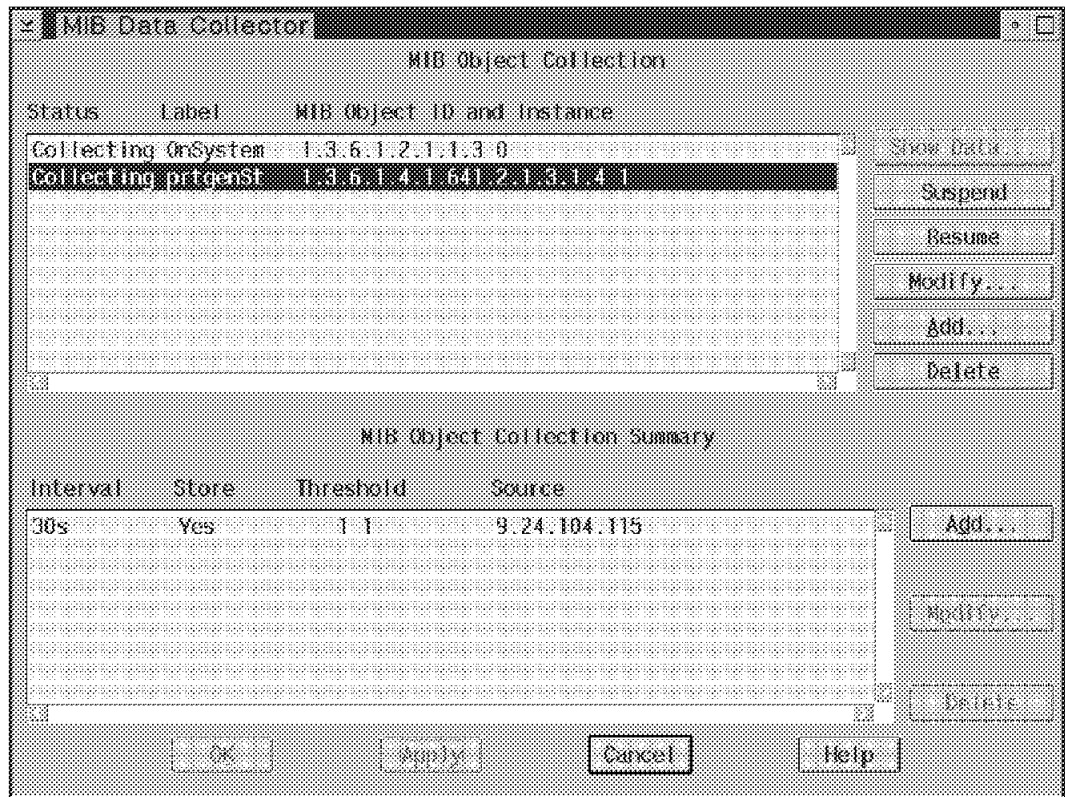


Figure 239. Collecting InputEmpty Data Values

NOTE

We must click on the **Apply** button for the Data Collector to actually start collecting data. Once you have clicked on it, the **Show Data...** button will be grayed out for at least as long as your polling interval. In our case it was 30 seconds before it had any data. This button only becomes active when there is some data to show.

When the Show Data button became active, we clicked on it and let the application run for a number of minutes. During this time, the printer was very busy, and it finally ran out of paper. We can see from the *ShowData: prtgenSt* panel as shown in Figure 240 on page 257 where the value of *prtgenStatusInputEmpty* goes from 1 to 2, and then back to 1 again when more paper has been added to the input tray.

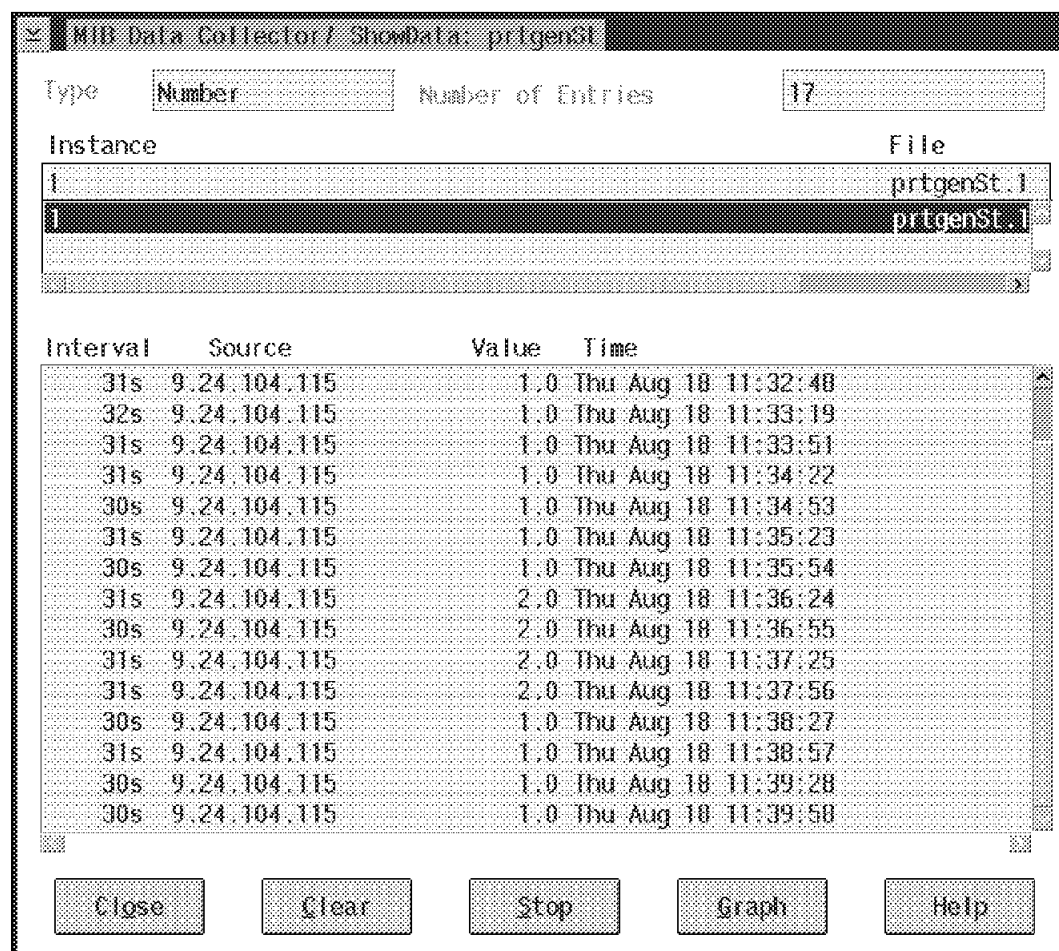


Figure 240. Showing InputEmpty Data Collected Over Time

11.4 Generating an Alert When the Status Changes

Since we used the Data Collector to set a threshold when `prtgenStatusInputEmpty` changed from a 1 to a 2, we will get an alert when this event occurs. The event can be displayed better in a graph. While you are viewing the values collected on the *ShowData: prtgenSt* panel as shown in Figure 240, click on the **Graph** button and you will get a graph of the values plotted over time as shown in Figure 241 on page 258.

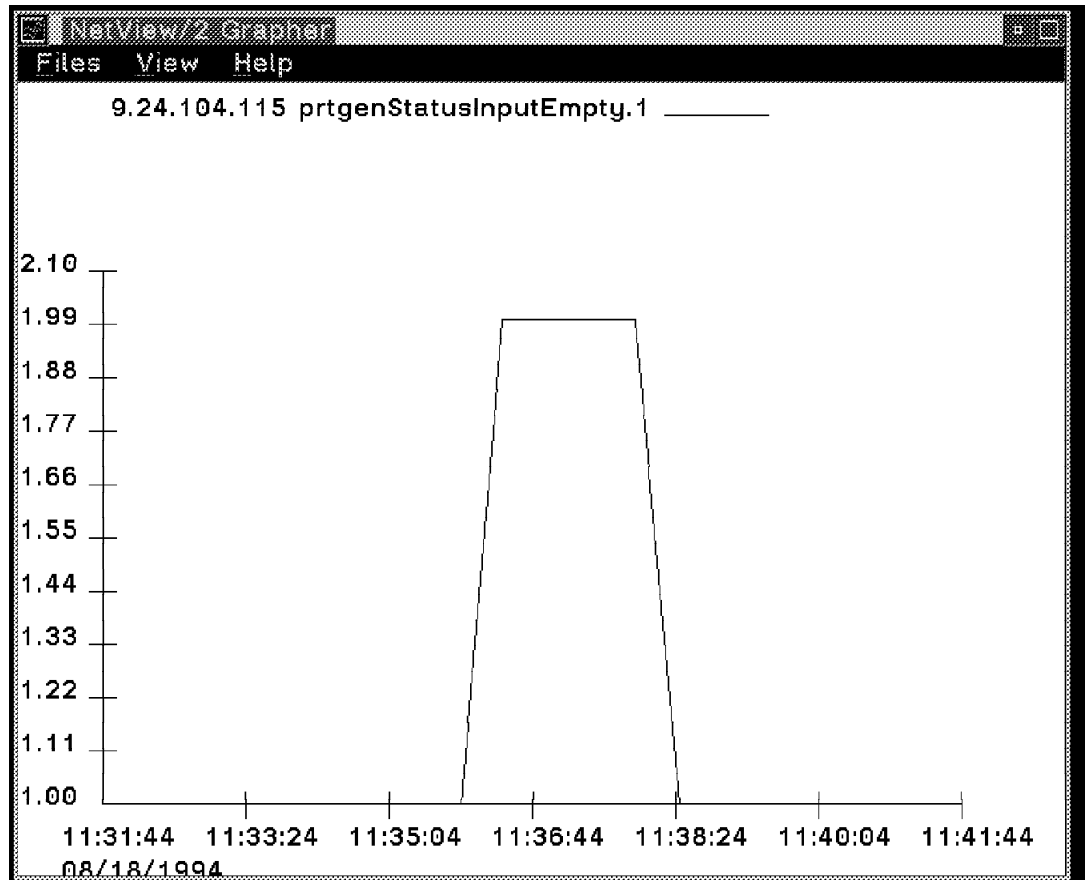


Figure 241. Graph Showing Where StatusInputEmpty Generated Alert

As you can see from the graph, the alert was generated at approximately 11:36 a.m. and the situation was back to normal at approximately 11:38 a.m.

11.5 Showing the Alert on the Event Displayer

In addition to showing the alert in graphical form, the alert is also sent as text to the managing station's trap log. The trap log can be viewed by selecting the **Event Displayer** from the NetView for OS/2 folder. If we scroll to the bottom of the event log, or limit what is displayed by choosing to show only threshold events, we can see the alerts generated by prtgenStatusInputEmpty changing from a 1 to a 2. Notice the bottom two lines of the Event Displayer window shown in Figure 242 on page 259.

Threshold Events			
	Node	Generic Specific	Description
13 14:58:40 1994 9.24.104.68	6	58720263	slaProcessorUtilizationBusy: 3-36, it exceeds the threshold 30
15 15:03:54 1994 nv2mgr2	6	58720263	snmpInPkts: 0-62, it exceeds the threshold 60
17 10:13:45 1994 9.24.104.115	6	50720263	prtgcnStatusInputEmpty: 1-2, it exceeds the threshold 1
17 10:15:46 1994 9.24.104.115	6	50720264	prtgcnStatusInputEmpty: 1-1, it is below the rearm-value 1
18 01:03:03 1994 9.24.104.54	6	58720263	sysUpTime: 0-3.6826e+06, it exceeds the threshold 3.6e+06
18 01:32:59 1994 9.24.104.42	6	58720263	sysUpTime: 0-3.6839e+06, it exceeds the threshold 3.6e+06
19 00:49:26 1994 9.24.104.42	6	50720264	sysUpTime: 0-61392, it is below the rearm-value 2.7e+06
18 11:36:55 1994 9.24.104.115	6	58720263	prtgcnStatusInputEmpty: 1-2, it exceeds the threshold 1
18 11:38:57 1994 9.24.104.115	6	50720264	prtgcnStatusInputEmpty: 1-1, it is below the rearm-value 1

Figure 242. Event Displayer Showing StatusInputEmpty Alert

Chapter 12. Scenario 2 - Monitoring LAN Server Generic Alerts

The objective of this chapter is to show the reader how NetView for OS/2 can trap on generic alerts that are forwarded from an OS/2 LAN Server. In this scenario, we will also show how NetView for OS/2 can automatically recover a failed resource on an OS/2 LAN Server. Specifically, we will be monitoring the following events:

- Error Log Limit Reached (generic alert 3D22B507).
- Authentication Failure (N bad password attempts - generic alert EF916293).

When these generic alerts come to the NetView for OS/2 managing station, we will use the facilities of the Event Automator to perform some automated recovery actions.

In the event that we receive the *Error Log Limit Reached* alert, we will:

- Display the system-supplied pop-up window with the appropriate error message on the managing system console.
- Use the facilities provided by LMU to initiate a REXX command file on the failed machine to print off the error log.
- Erase the error log so that we do not encounter the alert until it fills up again.

In the event that we receive the *Authentication Failure* alert, we will display the system-supplied pop-up window with the appropriate error message on the managing system console and then initiate a REXX command on the managing station that will do the following:

- Use the facilities of LMU to issue a remote command to the server in question. This command will be an OS/2 LAN Server command that will put a copy of the audit log onto a shared drive.
- We will then parse this audit log to extract *who* has been trying to log on with an invalid password.
- We will use the facilities of LMU to send an LMUPOPUP to the managing system console, identifying the user ID that was trying to repeatedly log on with an invalid password.

12.1 Monitoring for Maximum Error Log Size Exceeded

In order for our NetView for OS/2 managing system to monitor events on our remote OS/2 LAN Server, a number of things need to be running on that machine. Also, there are various settings that need to be made to a number of control files. This information will be presented in the following section.

12.1.1 Prerequisites for the Remote LAN Server

Following is a sequence of steps that you must take to set up your remote OS/2 LAN Server. Most of the changes involve the \IBMLAN\IBMLAN.INI file. A listing of the remote server's IBMLAN.INI file is given at the end of this section in 12.1.5, "IBMLAN.INI File for Remote OS/2 LAN Server" on page 270.

1. *Maximum Error Log Size* is a parameter that can be set to your specifications in your server's IBMLAN.INI file. The default for the

maxerrorlog is 100KB. In order for us to test this scenario, we brought the value down to 2KB as shown in the following extract from our IBMLAN.INI file:

```
maxerrorlog = 2
```

2. *alertnames* is another parameter that should be set in your server's IBMLAN.INI file. It should be set to the computername or user ID of the machine or person that will receive the pop-up message that is generated by LAN Server when the maximum error log size is reached. We set it so that it comes up on the server where this error situation occurs:

```
alertnames = NVSRV30
```

3. You must ensure that the following two LAN Server services are started when LAN Server starts:

- a. GENALERT
- b. ALERTER

12.1.2 Setting Up the Event Automator

This section will take the reader through the steps necessary to set up the NetView for OS/2 Event Automator in order to trap on the specific *Maximum Error Log Size Exceeded* trap, and initiate some actions based on that alert.

From the NetView for OS/2 Main Icon View window, double click on the **Event Automation** icon as shown in Figure 243.

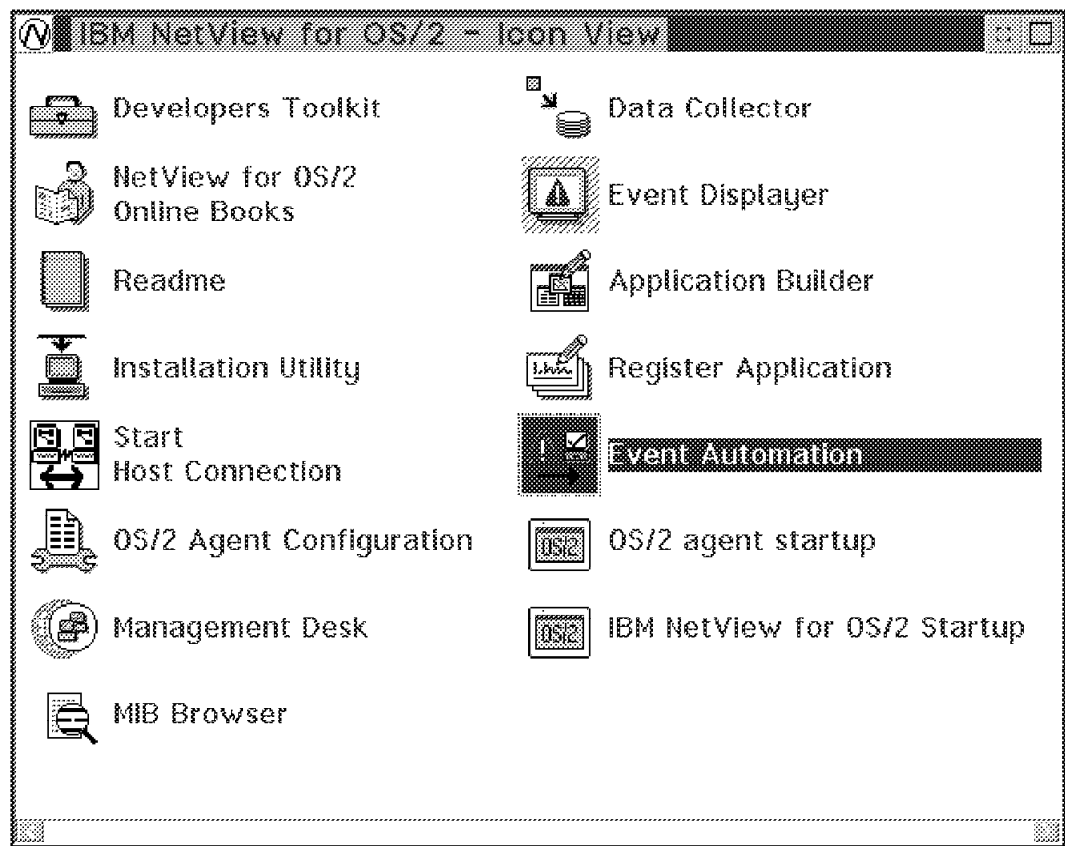


Figure 243. NetView for OS/2 Main Icon View

This will take you to the *Event Automation Update* main window. As you can see it is completely empty and waiting for input as shown in Figure 244 on page 263.

Event Automation Update

Event identification

Enterprise name	Enterprise ID	Generic trap	Specific trap

Add... Delete Add... Delete

Action specification

☒ Popup message

☒ Pager

Alias Message

☒ Forwarding

Address

☒ Status

New status

Optional command(s):

Find...

OK Apply Reset Cancel Help

Figure 244. Event Automation Update Window

The first thing we need to do is identify which event we want to trap on. In the *Event identification* subwindow there are four list boxes. We start at the left, and work our way across.

1. Click on the *Add...* button under the *Enterprise name* list box. This will present us with the *Event Automation Update - Add Enterprise* window as shown in Figure 245 on page 264.

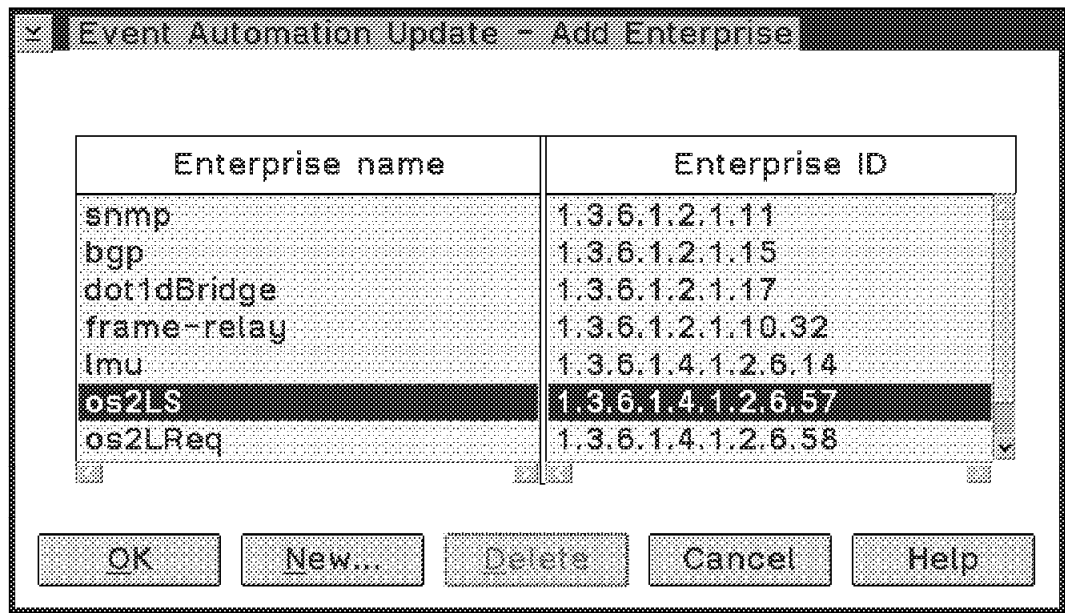


Figure 245. Event Automation - Add Enterprise Name and ID

2. Since we want to be monitoring for OS/2 LAN Server events, click on the *os2LS* line item in the *Enterprise name* list box and then click on the *OK* button. This will add the Enterprise Name and ID to *Enterprise name and ID* list boxes as shown in Figure 246 on page 265.
3. We must now specify which Generic and Specific Trap types we want to capture. To do this, we must click on the **os2LS** line item that we just added to the *Enterprise name* list box and then click on the *Add...* button under the *Generic Trap* list box. This will present us with the *Event Automation - Add Specific Traps* window as shown in the inset of Figure 246 on page 265.

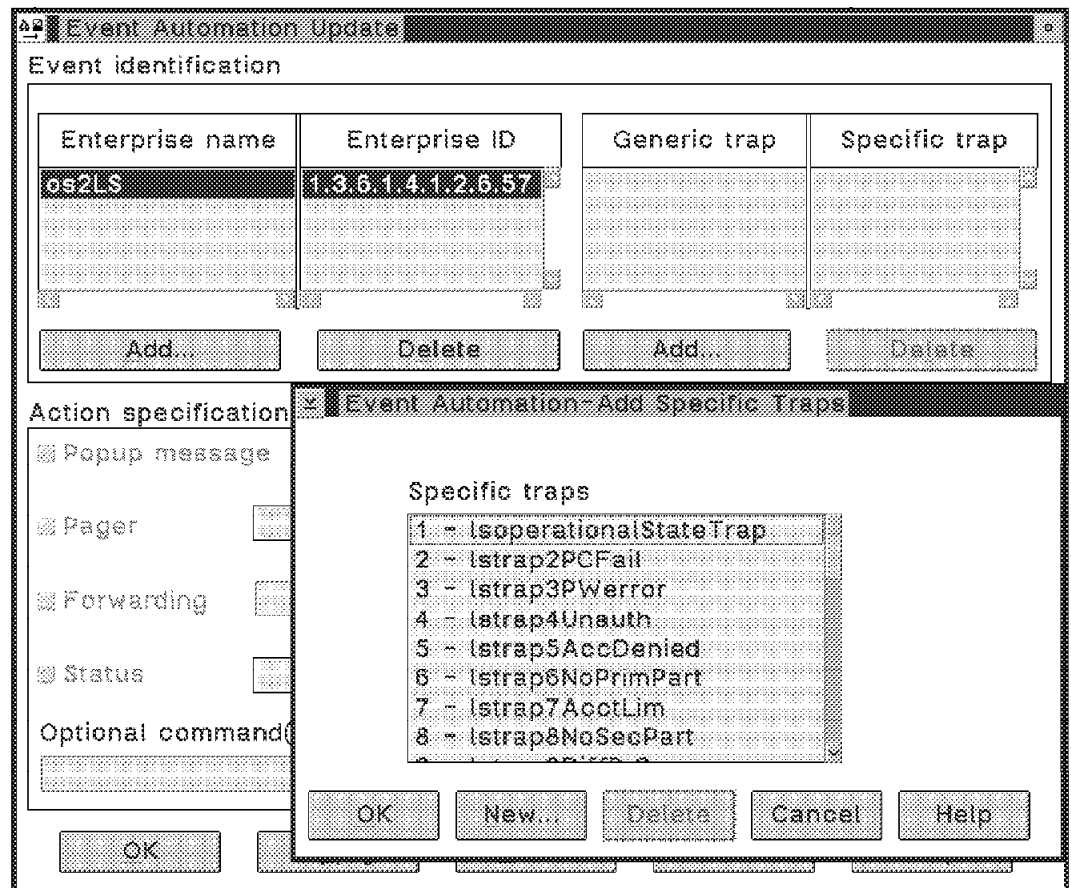


Figure 246. Event Automation - Adding Specific Trap Information

- We must now locate the *Maximum Error Log Size Exceeded* trap. Scroll down this list box to trap number 32, click on the 32 - *Istrap32MaxErr* line item and then click on the *OK* button as shown in Figure 247 on page 266.

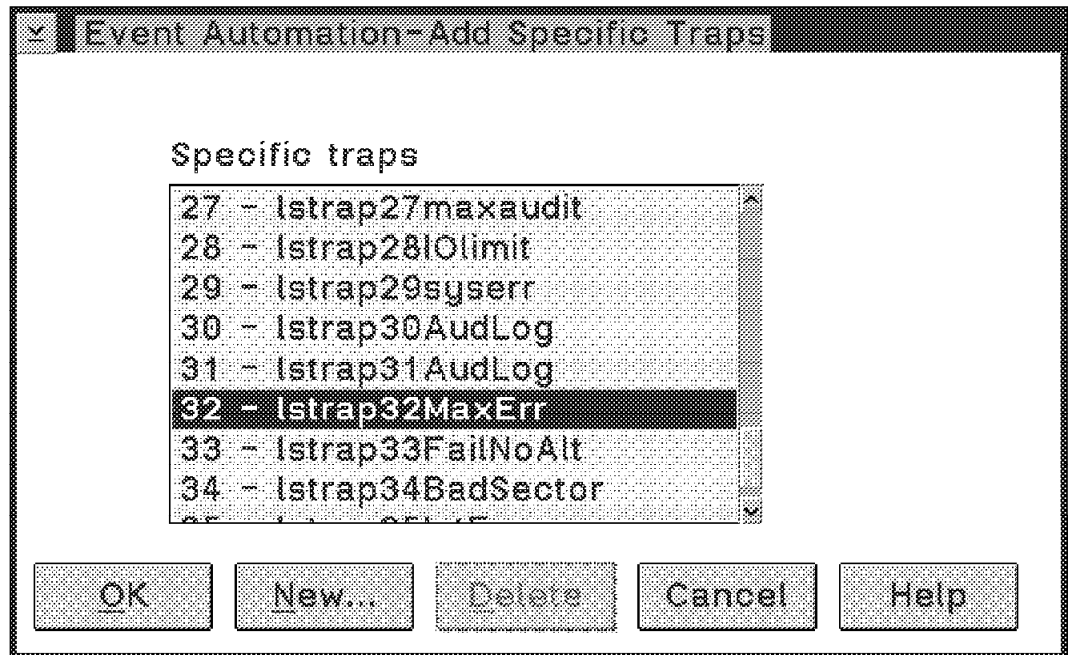


Figure 247. Event Automation - Adding the Istrap32MaxErr Trap

5. We now select the Generic/Specific Trap that we have just entered in the *Generic/Specific trap* list boxes as shown in Figure 248 on page 267. This will allow all the check boxes in the *Action specification* area to become active.
6. We would like two things to happen when this alert comes in.
 - a. Click on the *pop-up message* check box so that we get the system-supplied pop-up message.
 - b. Go to the *Optional command(s)* entry field and type `LMUCMD NVSRV30 D:\LMU2\FIXERRLG.CMD` as shown in Figure 248 on page 267. This will cause the FIXERRLG routine to be run on the NVSRV30 machine.

Note: You will have to substitute the name of your remote LAN Server machine in this command.

Event Automation Update

Event identification

Enterprise name	Enterprise ID	Generic trap	Specific trap
os2LS	1.3.6.1.4.1.2.6.57	6 - enterpriseS	32 - lstraps2M

Add... Delete Add... Delete

Action specification

☒ Popup message

☐ Pager Alias Message

☐ Forwarding Address

☐ Status New status

Optional command(s):

LMUCMD NVSRV30 D:\LMU2\FIXERRLG.CMD Find...

OK Apply Reset Cancel Help

Figure 248. Event Automation - Adding the Automation Actions

- Once all our actions have been entered into the preceding window, we can click on the *OK* button to save our updates and close the window.

Since we are using the utilities of LMU, it would be helpful to see the small network that we have set up. The LMU managing station is represented by the Star icon having address 00000009:400000033372, and our remote server is just to the right of it with a *computername* of NVSRV30 as shown in Figure 249 on page 268.

The LMU managing station was also the NetView for OS/2 managing system that we have just set up Event Automation on.

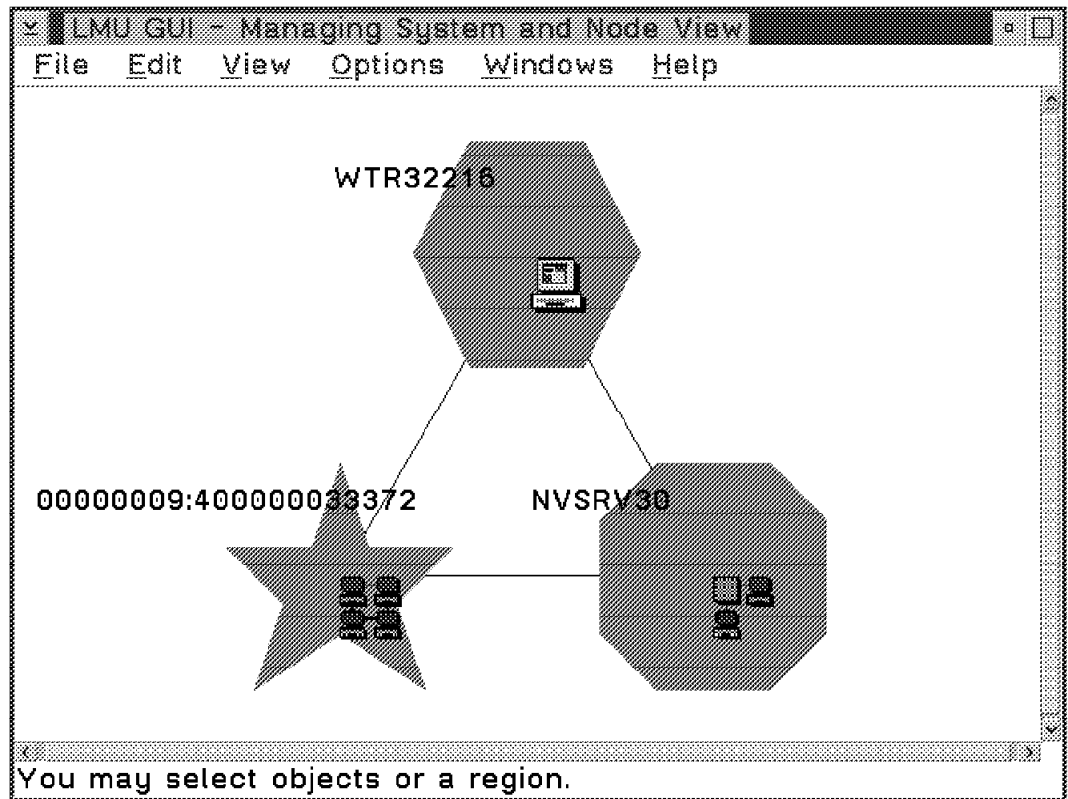


Figure 249. LMU Network Showing Managing and Managed Stations

12.1.3 Generating the Error at a Remote Server

We are now ready to start forcing errors into the remote server's error log. The error log is stored in the following file:

```
\ibmlan\logs\NET.ERR
```

If you go to this directory and type the *dir* command, you can see the size, in bytes, of this file. In order to force errors to be written to this file until it exceeds 2KB, we performed the following steps:

1. erase NET.ERR (to start from an empty file)
2. NET STOP SERVER
3. NET ACCOUNTS /ROLE:MEMBER (change role from Primary to Member)
4. NET START SERVER (this caused the Error and wrote to the Log)
5. NET ACCOUNTS /ROLE:PRIMARY (change role back to Primary)
6. NET START SERVER (everything is back to normal)
7. GOTO step 2

When the error log exceeds 2KB and we try to LOG ON to our Remote server, it will generate the following message:

```
NET3006E: The error log is full. No errors will be logged
until the file is cleared or the limit is raised.
```

and LAN Server will issue its own full-screen message pop-up to the machine that was identified in the *alertnames* field in the IBMLAN.INI file.

When the alert is generated, it is sent to the NetView for OS/2 managing station where it can be seen in the Event Displayer window as shown in Figure 250 on page 269.

Time	Node	Generic	Specific	Description
Aug 26 17:18:40 1994 9.24.104.54		6		Authentication failure trap: Inco...
Aug 26 17:30:54 1994 9.24.104.55		6	32	Enterprise: (.iso.org.dod.internet.private.ar chitecture.alert-product-Set-ID.1 fficinfoTable.hmfProductEntry.pro 1-3)(.iso.org.dod.internet.priv (.iso.org.dod.internet.private.ar chitecture.alert-product-Set-ID.1 fficinfoTable.hmfProductEntry.mad (.iso.org.dod.internet.private.ar

Figure 250. Event Displayer Window Showing LAN Server Alert #32

Notice in the above window that the alert has generic trap number 6 and specific trap number 32. That is what we had previously set up in the Event Automation windows, so this alert will now trigger two automated actions. The first automated action was to present the NetView for OS/2 console with the system-supplied pop-up window which is shown in Figure 251.

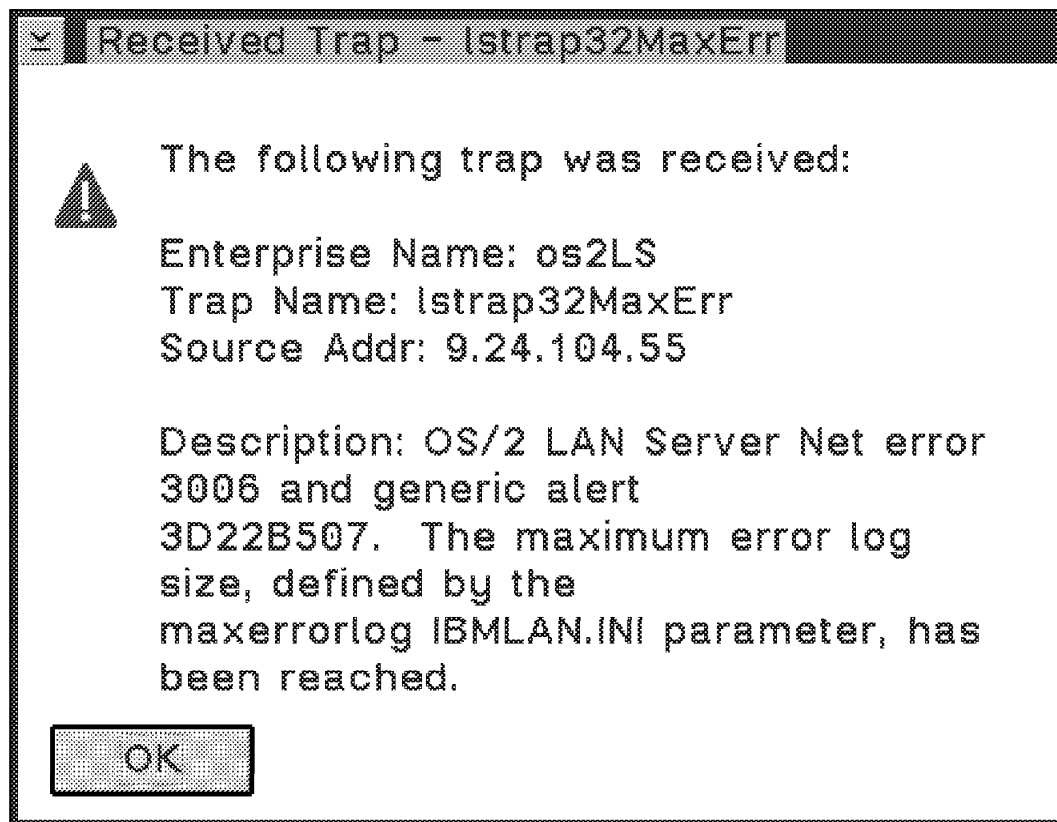


Figure 251. Event Automation - Pop-Up Showing Trap Received: lstrap32MaxErr

12.1.4 Resolving the Alert Situation

The second automated action was to use LMU to send a remote command to our server that would print the Error Log and then erase it. The REXX procedure to do this resides on the remote server as shown in Figure 252.

```
/* REXX Exec - FIXERRLG.CMD */
/* Author: Dan Heimann */
/*      ITS0 Raleigh */
/* Purpose: Dump OS/2 LAN Server Error Log */
/*      File (NET.ERR) to a printable file, */
/*      print it, and then clear it. */
/* */
TRACE OFF
'net error /r > d:\lmu2\neterror.log'
'copy d:\lmu2\neterror.log lpt2'
'erase d:\ibmlan\logs\net.err'
EXIT
```

Figure 252. FIXERRLG REXX Command File to Resolve MAXERRORLOG Alert

This REXX command file uses the OS/2 LAN Server *NET ERROR* command to put the contents of the NET.ERR file into a readable file in reverse order (*/r* parameter) so that it can be printed. Our printer had an alias of *lpt2*. You will have to change this REXX command file to use your printer alias.

The next time something is written to the LAN Server error log, it will recreate a new NET.ERR file.

12.1.5 IBMLAN.INI File for Remote OS/ LAN Server

; OS/2 LAN Server initialization file

networks

net1 = NETBEUI\$,0,LM10,32,50,14

; This information is read by the redirector at device initialization time.

requester

COMPUTERNAME = NVSRV30

DOMAIN = NVSRVDM

; The following parameters generally do not need to be
; changed by the user.

charcount = 16

chartime = 250

charwait = 3600

keepconn = 600

keepsearch = 600

maxcmds = 16

maxerrorlog = 2

maxthreads = 10

maxwrkcache = 64

numalerts = 12

numcharbuf = 10

numservices = 16

numworkbuf = 15

numdgrambuf = 14

othdomains =

```

printbuftime = 90
sesstimeout = 45
sizcharbuf = 512
sizerror = 1024
sizworkbuf = 4096
; The next lines help you to locate bits in the wrkheuristics entry.
;
;           1       2       3
;           0123456789012345678901234567890123
wrkheuristics = 1111111121311111110001011120111221
WRKSERVICES = LSCLIENT, MESSENGER, NETPOPOP
wrknets = NET1

messenger

logfile = messages.log
sizmessbuf = 4096

lsclient

multilogon = no
timesync = yes
logonverification = domain
logonwarningmsgs = all

netlogon

SCRIPTS = D:\IBMLAN\REPL\IMPORT\SCRIPTS
pulse = 60
update = yes

replicator

replicate = IMPORT
IMPORTPATH = D:\ibmlan\repl\import
tryuser = yes
password =
interval = 5
guardtime = 2
pulse = 3
random = 60

dcdbrep1

tryuser = yes
password =
interval = 5
guardtime = 2
pulse = 3
random = 60

server

alertnames = NVSRV30
auditing = yes
autodisconnect = 120
maxusers = 32
; The following parameters generally do not need to be
; changed by the user. NOTE: srvnets= is represented in

```

```

; the server info struct as a 16-bit lan mask. Srvnet names
; are converted to indexes within networks for the named nets.
  guestacct = guest
  accessalert = 5
  alertsched = 5
  diskalert = 10000
  erroralert = 5
  logonalert = 5
  maxauditlog = 100
  maxchdevjob = 6
  maxchdevq = 2
  maxchdevs = 2
  maxconnections = 128
  maxlocks = 64
  maxopens = 250
  maxsearches = 50
  maxsessopens = 80
  maxsessreqs = 50
  maxsessvcs = 1
  maxshares = 16
  netioalert = 5
  numbigbuf = 12
  numfiletasks = 1
  numreqbuf = 36
  sizreqbuf = 4096
  srvanndelta = 3000
  srvannounce = 60
; The next lines help you to locate bits in the srvheuristics entry.
;
;                               1
;                               01234567890123456789
  srvheuristics = 11110141111311001331
  SRVSERVICES = NETLOGON,LSSERVER,GENALERT,ALERter
  srvnets = NET1

alerter

  sizalertbuf = 3072

netrun

  maxruns = 3
  runpath = C:\

lsserver

  cleanup = yes
  srvpipes = 3

services

; Correlates name of service to pathname of service program.
; The pathname must be either
;     1) an absolute path (including the drive specification)
;     OR
;     2) a path relative to the IBMLAN root
  alerter = services\alerter.exe
  dcdbrepl = services\dcdbrepl.exe
  dlrintst = services\dlrintst.exe

```

```

genalert = services\genalert.exe
; Correlates name of service to pathname of service program.
; The pathname must be either
;     1) an absolute path (including the drive specification)
;         OR
;     2) a path relative to the IBMLAN root
lsclient = services\lsclient.exe
lsserver = services\lsserver.exe
messenger = services\msrvinit.exe
netlogon = services\netlogon.exe
netpopup = services\netpopup.exe
netrun = services\runservr.exe
remoteboot = services\rplservr.exe
replicator = services\replicat.exe
requester = services\wksta.exe
server = services\netsvini.exe
timesource = services\timesrc.exe
ups = services\ups.exe

```

Appendix A. LMU/2 Related Files

A.1 LMU/2 Control File (LMU.CTL) for the Managing System

```
#####
# IDENTIFIES THE PATH AND FILENAME OF THE LMU PROFILE.      #
# THE LMU PROFILE MUST RESIDE ON THE OS/2 BOOT DRIVE AND    #
# HAVE THE NAME LMU.INI                                     #
#####

DEFINE_PROFILE INI_FILE(C:\LMU.INI)

#####
#
# THE FOLLOWING PARAMETERS DO NOT HAVE A DEFAULT VALUE      #
# AND MUST BE MODIFIED BEFORE USING THE COMPONENT WHICH     #
# REFERENCES THEM:                                           #
#
#   MANAGING_SYSTEM                                           #
#   MANAGING_SYSTEM_WITH_DATABASE                           #
#   FAULT_MANAGER                                             #
#   GRAPHICAL_USER_INTERFACE                                 #
#   SNMP_PROXY_AGENT                                          #
#   SNMP_PROXY_INFORMATION                                   #
#
#####

#####
# THE FOLLOWING PARAMETERS APPLY TO ALL WORKSTATIONS.        #
#####

# The computername or internetwork address specified
# identifies this workstation's managing system.
# Ex. LMUMANG (IBM requester)
#      or
# Ex. 000000A1:100012345678 (NetWare requester)

APP(LMU_UTILITY),
   KEY(MANAGING_SYSTEM),
   ASCIIZ(00000009:400000033372);

# The computername or internetwork address specified
# identifies this workstation's managing system with database,
# which is the system maintaining the LMU database.
# Ex. LMUMANG (IBM requester)
#      or
# Ex. 000000A1:100012345678 (NetWare requester)

APP(LMU_UTILITY),
   KEY(MANAGING_SYSTEM_WITH_DATABASE),
   ASCIIZ(00000009:400000033372);

# The computername or internetwork address specified identifies this
# workstation's fault manager, which is the system to receive alerts
```

```

# generated by the LMU applications.
# Ex. FAULTMAN (IBM requester)
#      or
# Ex. 000000A1:100012345678 (NetWare requester)

APP(LMU_UTILITY),
    KEY(FAULT_MANAGER),
    ASCIIIZ(00000009:400000033372);

# Identifies the location of the file to contain
# the messages issued by LMU.

APP(LMU_UTILITY),
    KEY(MESSAGE_LOG),
    ASCIIIZ(D:\LMU2\LMU.LOG);

# Identifies the LAN adapter used in NETBIOS communications.
# Value 00 indicates the primary adapter and value 01 indicates
# the secondary adapter. This key is optional and if not
# specified the primary adapter (00) will be used.
#
# NOTE: The hexnum value for LAN_ADAPTER must be specified
# as 2 hexadecimal digits (for example, 01).

APP(LMU_UTILITY),
    KEY(LAN_ADAPTER),
    HNUM(00);

# Identifies the location in which the *.BND files were
# installed.
# Managing System and SNMP Proxy Agent workstations using
# database ONLY.

APP(LMU_UTILITY),
    KEY(BIND),
    ASCIIIZ(D:\LMU2);

#####
#   THE FOLLOWING PARAMETERS APPLY TO "MANAGED SYSTEMS".   #
#####

# Identifies the frequency in minutes that the heartbeat
# function will send a message to the managing system.
#
# NOTE: The hexnum value must be specified as 4 hexadecimal
#       digits, e.g. (000A) to indicate 10 minutes.
#
# If '0000' is specified only the initial and terminal
# heartbeats are sent.

APP(LMU_UTILITY),
    KEY(PULSE_RATE),
    HNUM(0001);

```



```

#####
# THE FOLLOWING PARAMETERS APPLY TO "MANAGING SYSTEMS".      #
#####

# Identifies the file to contain
# the node description change log.

APP(LMU_UTILITY),
    KEY(CHANGE_LOG),
    ASCIIIZ(D:\LMU2\CHANGE.LOG);

# Identifies the location to which transferred
# files are to written.

APP(LMU_UTILITY),
    KEY(FILE_PATH),
    ASCIIIZ(D:\LMU2);

#####
# THE FOLLOWING PARAMETERS APPLY TO "FAULT MANAGER"          #
# WORKSTATIONS.                                              #
#####

# Identifies the Fault Manager's input user table.
# For example D:\LMU2\AUEUSER.TAB

APP(LMU_UTILITY),
    KEY(FAULT_TABLE),
    ASCIIIZ(D:\LMU2\AUEUSER.TAB);

# Alerts can be forwarded to a specific adapter address if desired.
# This key is optional and if not specified the default LAN management
# functional address of 'C00000002000' is used.

APP(LMU_UTILITY),
    KEY(FM_FORWARDING_ADDR),
    ASCIIIZ(C00000002000);

# Identifies the computer names and/or internetwork addresses
# of the GUI workstations to the FAULT MANAGER.
#
# Note: The character values for this field must be specified as
# character fields separated by a comma.
# Ex. (LMUGUI,000000A1:100012345678)

APP(LMU_UTILITY),
    KEY(GRAPHICAL_USER_INTERFACE),
    ASCIIIZ(00000009:400000033372);

# Identifies the computer names and/or internetwork addresses
# of the SNMP Proxy Agent workstations to the FAULT MANAGER.
#
# Note: The character values for this field must be specified as
# character fields separated by a comma.
# Ex. (LMUSNMPD,000000A1:100012345678)

```

```

APP(LMU_UTILITY),
    KEY(SNMP_PROXY_AGENT),
    ASCIIIZ(00000009:40000033372);

#####
# THE FOLLOWING PARAMETERS APPLY TO "SCHEDULER" #
# WORKSTATIONS. #
#####

# Identifies the path and file name of the Schedule log file
# For example D:\LMU2\SCHEDULE.LOG

APP(LMU_UTILITY),
    KEY(SCHEDULE_LOG),
    ASCIIIZ(D:\LMU2\SCHEDULE.LOG);

# Identifies the path and file name of the Schedule file
# For example D:\LMU2\SCHEDULE.TIM

APP(LMU_UTILITY),
    KEY(SCHEDULE_FILE),
    ASCIIIZ(D:\LMU2\SCHEDULE.TIM);

# Identifies the path and file name of the Schedule group file
# For example D:\LMU2\SCHEDULE.GRP

APP(LMU_UTILITY),
    KEY(SCHEDULE_GROUP_FILE),
    ASCIIIZ(D:\LMU2\SCHEDULE.GRP);

# Identifies the frequency in minutes that the schedule
# file will be checked for changes.
# For example 60

APP(LMU_UTILITY),
    KEY(SCHEDULE_READ),
    ASCIIIZ(60);

#####
# THE FOLLOWING PARAMETERS APPLY TO "ADMINISTRATOR" #
# WORKSTATIONS RUNNING THE GRAPHICAL USER INTERFACE (GUI). #
#####

# Indicates additional managing systems to be queried
# at GUI startup.
#
# Note: The character values for this field must be specified as
# character fields separated by a comma.
# For example: (LmuMgr,HelpDesk,Ops1).

#APP(LMU_UTILITY),
# KEY(GUI_ADDITIONAL_MANAGING_SYSTEMS),
# ASCIIIZ(computername or internetwork address,...);

```

```

# Indicates the type of view to be displayed at GUI startup.
#
# For additional information see GUI documentation.

APP(LMU_UTILITY),
    KEY(GUI_INITIAL_DISPLAY),
    ASCIIZ(A);

# Indicates which symbol is associated with the thirteen types of
# view objects.
#
# NOTE: The hexnum value for this field must be specified
# as 26 hexadecimal digits, e.g. (04030A0806020709050B01120E).
# For additional information see GUI documentation.

APP(LMU_UTILITY),
    KEY(GUI_NODE_SYMBOLS),
    HNUM(04030A0806020709050B01120E);

# Indicates the text color and background color to be used
# when commands are submitted via the GUI.
#
# Note: The character values for this field must be specified
# as two 2-digit numbers separated by a comma, e.g. (34,47)
# to indicate white text on a black background.
# For additional information see GUI documentation.

APP(LMU_UTILITY),
    KEY(GUI_COLORS),
    ASCIIZ(34,47);

# Indicates the name and the nominal point size, in tenths, of the
# image font used to display all text in the GUI window.
#
# Note: The character values for this field must be specified as a
# character field and a 3-digit number separated by a comma,
# e.g. (Helv,100) to indicate Helvetica 10 point.
# For additional information see GUI documentation.

#APP(LMU_UTILITY),
#    KEY(GUI_FONTS),
#    ASCIIZ(System Proportional,120);

# Identifies the location of the file to contain
# command sequences store by the GUI.

APP(LMU_UTILITY),
    KEY(GUI_COMMANDS_TABLE),
    ASCIIZ(D:\LMU2\LMUGUI.TAB);

# Specifies which pattern is used to indicate that a node
# has received alerts or that a collection has subordinate
# nodes that have received alerts.
#

```

```

# NOTE: The hexnum value for this field must be specified
# as 2 hexadecimal digits, e.g. (0C) to indicate medium
# density, diagonal hash marks.
# For additional information see GUI documentation.

APP(LMU_UTILITY),
    KEY(GUI_PATTERN),
    HNUM(0C);

# Indicates if the view is to be refreshed when a resource is
# added due to the arrival of an event for an unknown workstation
# that matches the type of resources being displayed.
# Coding N or n indicates that an automatic refresh is not done.
#
# For additional information see GUI documentation.

APP(LMU_UTILITY),
    KEY(GUI_AUTO_REFRESH),
    ASCIIIZ(Y);

# Indicates the height and width ratios that will be used as the
# aspect ratio for the ellipse used to display the workstations.
#
# Note: The character values for this field must be specified as
# two 3-digit numbers separated by a comma, e.g. (480,640) to
# indicate an ellipse that approximates the height to width ratio
# of a standard file monitor in 640 X 480 mode.
# Note: specifying (001,001) will result in a circle.
# For additional information see GUI documentation.

APP(LMU_UTILITY),
    KEY(GUI_COORDINATES),
    ASCIIIZ(001,001);

# Indicates the resource layout limit to be used when displaying
# the GUI views.
#
# Note: The character value for this field must be specified as
# a 3-digit number. For example: (020) to indicate that up to
# 20 resources will be displayed as an ellipse, while 21 or more
# will be displayed in rows and columns.
# For additional information see the GUI documentation.

APP(LMU_UTILITY),
    KEY(GUI_RESOURCE_LAYOUT_LIMIT),
    ASCIIIZ(020);

#####
#   THE FOLLOWING PARAMETERS APPLY TO "SNMP PROXY AGENT"   #
#   WORKSTATIONS.                                           #
#####

# Indicates additional managing systems to be queried
# at SNMP Proxy Agent startup.
#
# Note: The character values for this field must be specified as

```

```

# character fields separated by a comma.
# For example: (LmuMgr,HelpDesk,Ops1).

#APP(LMU_UTILITY),
#   KEY(PROXY_ADDITIONAL_MANAGING_SYSTEMS),
#   ASCIIZ(computername or internetwork address,...);

#####
#   THE FOLLOWING PARAMETERS APPLY TO "LAN NETVIEW" (LMULNV)   #
#   WORKSTATIONS.                                             #
#####

# Identifies the LMU SNMP Proxy Agents to be queried at LMULNV
# startup. Specify the IP address (or host name) along with
# a community name for each proxy agent. The community name
# defaults to "public" if not specified.
#
# For example: (.LmuProxy,public',.9.179.7.66,rtp',.9.179.7.50,').

APP(LMU_UTILITY),
    KEY(SNMP_PROXY_INFORMATION),
    ASCIIZ(.nv2mgr1,public');

#####
#   THE FOLLOWING PARAMETERS APPLY TO THE APPWATCH UTILITY.   #
#####

# Identifies the path and file name of the application watch table.
# For example D:\LMU2\APPWATCH.SMP

APP(LMU_UTILITY),
    KEY(APPWATCH_TABLE),
    ASCIIZ(D:\LMU2\APPWATCH.TAB);

```

A.2 USERVPD.CFG File

```
1) Assigned_user           Heimann, Dan
2) User Serial #           025852
3) User department #       002
4) User Internal Phone #   316-2949
5) User External Phone #   416-462-2576
6) Building                3600
7) Floor                   OE5
8) Location Office #       E501
9) Location Internal Phone # 352-2428
10) Location External Phone # 919-850-0293
11) Owning_Manager         Gaillour, Jose
12) Owning_Department      002
13) T/R Port ID            E5501B55FD
14) Power Outlet           E5501FF
15) Equipment
    Format: {Machine_Type,Model,Serial_Number,Date_Installed}
```

```
Start of equipment
    {IBM-8595,001,23-1234567,09-01-1992}    System Unit
    {IBM-8514,001,00-1097175,09-01-1992}    Display
    {IBM-8513,001,23-CLV46,09-01-1992}      Display
    {IBM-1391401,,4655517,09-01-1992}       Keyboard
    {IBM-90X6778,,1306518,09-01-1992}       Mouse
    {IBM-4216,031,41-8886A,09-01-1992}      Printer
    {IBM-6180,,B1506,09-01-1992}           Plotter
End of equipment
```

A.3 LMU/2 Control File (LMU.CTL) for the Managed System

```
#####
# IDENTIFIES THE PATH AND FILENAME OF THE LMU PROFILE. #
# THE LMU PROFILE MUST RESIDE ON THE OS/2 BOOT DRIVE AND #
# HAVE THE NAME LMU.INI #
#####

DEFINE_PROFILE INI_FILE(C:\LMU.INI)

#####
#
# THE FOLLOWING PARAMETERS DO NOT HAVE A DEFAULT VALUE #
# AND MUST BE MODIFIED BEFORE USING THE COMPONENT WHICH #
# REFERENCES THEM: #
#
# MANAGING_SYSTEM #
# MANAGING_SYSTEM_WITH_DATABASE #
# FAULT_MANAGER #
# GRAPHICAL_USER_INTERFACE #
# SNMP_PROXY_AGENT #
# SNMP_PROXY_INFORMATION #
#
#####

#####
# THE FOLLOWING PARAMETERS APPLY TO ALL WORKSTATIONS. #
#####

# The computername or internetwork address specified
# identifies this workstation's managing system.
# Ex. LMUMANG (IBM requester)
# or
# Ex. 000000A1:100012345678 (NetWare requester)

APP(LMU_UTILITY),
    KEY(MANAGING_SYSTEM),
    ASCIIZ(WTR33372);

# The computername or internetwork address specified
# identifies this workstation's managing system with database,
# which is the system maintaining the LMU database.
# Ex. LMUMANG (IBM requester)
# or
# Ex. 000000A1:100012345678 (NetWare requester)

APP(LMU_UTILITY),
    KEY(MANAGING_SYSTEM_WITH_DATABASE),
    ASCIIZ(WTR33372);

# The computername or internetwork address specified identifies this
# workstation's fault manager, which is the system to receive alerts
# generated by the LMU applications.
# Ex. FAULTMAN (IBM requester)
# or
# Ex. 000000A1:100012345678 (NetWare requester)
```

```

APP(LMU_UTILITY),
    KEY(FAULT_MANAGER),
    ASCIIZ(WTR33372);

# Identifies the location of the file to contain
# the messages issued by LMU.

APP(LMU_UTILITY),
    KEY(MESSAGE_LOG),
    ASCIIZ(D:\LMU2\LMU.LOG);

# Identifies the LAN adapter used in NETBIOS communications.
# Value 00 indicates the primary adapter and value 01 indicates
# the secondary adapter. This key is optional and if not
# specified the primary adapter (00) will be used.
#
# NOTE: The hexnum value for LAN_ADAPTER must be specified
# as 2 hexadecimal digits (for example, 01).

APP(LMU_UTILITY),
    KEY(LAN_ADAPTER),
    HNUM(00);

# Identifies the location in which the *.BND files were
# installed.
# Managing System and SNMP Proxy Agent workstations using
# database ONLY.

APP(LMU_UTILITY),
    KEY(BIND),
    ASCIIZ(D:\LMU2);

#####
# THE FOLLOWING PARAMETERS APPLY TO "MANAGED SYSTEMS".      #
#####

# Identifies the frequency in minutes that the heartbeat
# function will send a message to the managing system.
#
# NOTE: The hexnum value must be specified as 4 hexadecimal
#       digits, e.g. (000A) to indicate 10 minutes.
#
# If '0000' is specified only the initial and terminal
# heartbeats are sent.

APP(LMU_UTILITY),
    KEY(PULSE_RATE),
    HNUM(0001);

#####
# THE FOLLOWING PARAMETERS APPLY TO "MANAGING SYSTEMS".      #
#####

# Identifies the file to contain
# the node description change log.

```



```

APP(LMU_UTILITY),
    KEY(CHANGE_LOG),
    ASCIIIZ(D:\LMU2\CHANGE.LOG);

# Identifies the location to which transferred
# files are to written.

APP(LMU_UTILITY),
    KEY(FILE_PATH),
    ASCIIIZ(D:\LMU2);

#####
# THE FOLLOWING PARAMETERS APPLY TO "FAULT MANAGER" #
# WORKSTATIONS. #
#####

# Identifies the Fault Manager's input user table.
# For example D:\LMU2\AUEUSER.TAB

APP(LMU_UTILITY),
    KEY(FAULT_TABLE),
    ASCIIIZ(D:\LMU2\AUEUSER.TAB);

# Alerts can be forwarded to a specific adapter address if desired.
# This key is optional and if not specified the default LAN management
# functional address of 'C00000002000' is used.

APP(LMU_UTILITY),
    KEY(FM_FORWARDING_ADDR),
    ASCIIIZ(C00000002000);

# Identifies the computer names and/or internetwork addresses
# of the GUI workstations to the FAULT MANAGER.
#
# Note: The character values for this field must be specified as
# character fields separated by a comma.
# Ex. (LMUGUI,000000A1:100012345678)

APP(LMU_UTILITY),
    KEY(GRAPHICAL_USER_INTERFACE),
    ASCIIIZ(WTR33372);

# Identifies the computer names and/or internetwork addresses
# of the SNMP Proxy Agent workstations to the FAULT MANAGER.
#
# Note: The character values for this field must be specified as
# character fields separated by a comma.
# Ex. (LMUSNMPD,000000A1:100012345678)

APP(LMU_UTILITY),
    KEY(SNMP_PROXY_AGENT),
    ASCIIIZ(LMUSNMPD,00000009:400000033372);

#####
# THE FOLLOWING PARAMETERS APPLY TO "SCHEDULER" #

```

```

# WORKSTATIONS. #
#####

# Identifies the path and file name of the Schedule log file
# For example D:\LMU2\SCHEDULE.LOG

APP(LMU_UTILITY),
  KEY(SCHEDULE_LOG),
  ASCIIZ(D:\LMU2\SCHEDULE.LOG);

# Identifies the path and file name of the Schedule file
# For example D:\LMU2\SCHEDULE.TIM

APP(LMU_UTILITY),
  KEY(SCHEDULE_FILE),
  ASCIIZ(D:\LMU2\SCHEDULE.TIM);

# Identifies the path and file name of the Schedule group file
# For example D:\LMU2\SCHEDULE.GRP

APP(LMU_UTILITY),
  KEY(SCHEDULE_GROUP_FILE),
  ASCIIZ(D:\LMU2\SCHEDULE.GRP);

# Identifies the frequency in minutes that the schedule
# file will be checked for changes.
# For example 60

APP(LMU_UTILITY),
  KEY(SCHEDULE_READ),
  ASCIIZ(60);

#####
# THE FOLLOWING PARAMETERS APPLY TO "ADMINISTRATOR" #
# WORKSTATIONS RUNNING THE GRAPHICAL USER INTERFACE (GUI). #
#####

# Indicates additional managing systems to be queried
# at GUI startup.
#
# Note: The character values for this field must be specified as
# character fields separated by a comma.
# For example: (LmuMgr,HelpDesk,Ops1).

#APP(LMU_UTILITY),
#  KEY(GUI_ADDITIONAL_MANAGING_SYSTEMS),
#  ASCIIZ(computername or internetwork address,...);

# Indicates the type of view to be displayed at GUI startup.
#
# For additional information see GUI documentation.

APP(LMU_UTILITY),
  KEY(GUI_INITIAL_DISPLAY),
  ASCIIZ(A);

```

```
# Indicates which symbol is associated with the thirteen types of  
# view objects.
```

```
#  
# NOTE: The hexnum value for this field must be specified  
# as 26 hexadecimal digits, e.g. (04030A0806020709050B01120E).  
# For additional information see GUI documentation.
```

```
APP(LMU_UTILITY),  
    KEY(GUI_NODE_SYMBOLS),  
    HNUM(04030A0806020709050B01120E);
```

```
# Indicates the text color and background color to be used  
# when commands are submitted via the GUI.
```

```
#  
# Note: The character values for this field must be specified  
# as two 2-digit numbers separated by a comma, e.g. (34,47)  
# to indicate white text on a black background.  
# For additional information see GUI documentation.
```

```
APP(LMU_UTILITY),  
    KEY(GUI_COLORS),  
    ASCIIZ(34,47);
```

```
# Indicates the name and the nominal point size, in tenths, of the  
# image font used to display all text in the GUI window.
```

```
#  
# Note: The character values for this field must be specified as a  
# character field and a 3-digit number separated by a comma,  
# e.g. (Helv,100) to indicate Helvetica 10 point.  
# For additional information see GUI documentation.
```

```
#APP(LMU_UTILITY),  
#    KEY(GUI_FONTS),  
#    ASCIIZ(System Proportional,120);
```

```
# Identifies the location of the file to contain  
# command sequences store by the GUI.
```

```
APP(LMU_UTILITY),  
    KEY(GUI_COMMANDS_TABLE),  
    ASCIIZ(D:\LMU2\LMUGUI.TAB);
```

```
# Specifies which pattern is used to indicate that a node  
# has received alerts or that a collection has subordinate  
# nodes that have received alerts.
```

```
#  
# NOTE: The hexnum value for this field must be specified  
# as 2 hexadecimal digits, e.g. (0C) to indicate medium  
# density, diagonal hash marks.  
# For additional information see GUI documentation.
```

```
APP(LMU_UTILITY),  
    KEY(GUI_PATTERN),
```

```

HNUM(OC);

# Indicates if the view is to be refreshed when a resource is
# added due to the arrival of an event for an unknown workstation
# that matches the type of resources being displayed.
# Coding N or n indicates that an automatic refresh is not done.
#
# For additional information see GUI documentation.

APP(LMU_UTILITY),
  KEY(GUI_AUTO_REFRESH),
  ASCIIZ(Y);

# Indicates the height and width ratios that will be used as the
# aspect ratio for the ellipse used to display the workstations.
#
# Note: The character values for this field must be specified as
# two 3-digit numbers separated by a comma, e.g. (480,640) to
# indicate an ellipse that approximates the height to width ratio
# of a standard file monitor in 640 X 480 mode.
# Note: specifying (001,001) will result in a circle.
# For additional information see GUI documentation.

APP(LMU_UTILITY),
  KEY(GUI_COORDINATES),
  ASCIIZ(001,001);

# Indicates the resource layout limit to be used when displaying
# the GUI views.
#
# Note: The character value for this field must be specified as
# a 3-digit number. For example: (020) to indicate that up to
# 20 resources will be displayed as an ellipse, while 21 or more
# will be displayed in rows and columns.
# For additional information see the GUI documentation.

APP(LMU_UTILITY),
  KEY(GUI_RESOURCE_LAYOUT_LIMIT),
  ASCIIZ(020);

#####
# THE FOLLOWING PARAMETERS APPLY TO "SNMP PROXY AGENT"      #
# WORKSTATIONS.                                           #
#####

# Indicates additional managing systems to be queried
# at SNMP Proxy Agent startup.
#
# Note: The character values for this field must be specified as
# character fields separated by a comma.
# For example: (LmuMgr,HelpDesk,Ops1).

#APP(LMU_UTILITY),
#  KEY(PROXY_ADDITIONAL_MANAGING_SYSTEMS),
#  ASCIIZ(computername or internetwork address,...);

```

```
#####
#   THE FOLLOWING PARAMETERS APPLY TO "LAN NETVIEW" (LMULNV)   #
#   WORKSTATIONS.                                             #
#####

# Identifies the LMU SNMP Proxy Agents to be queried at LMULNV
# startup. Specify the IP address (or host name) along with
# a community name for each proxy agent. The community name
# defaults to "public" if not specified.
#
# For example: (.LmuProxy,public',.9.179.7.66,rtp',.9.179.7.50,').

APP(LMU_UTILITY),
    KEY(SNMP_PROXY_INFORMATION),
    ASCIIIZ(.nv2mgr1,public');

#####
#   THE FOLLOWING PARAMETERS APPLY TO THE APPWATCH UTILITY.   #
#####

# Identifies the path and file name of the application watch table.
# For example D:\LMU2\APPWATCH.SMP

APP(LMU_UTILITY),
    KEY(APPWATCH_TABLE),
    ASCIIIZ(D:\LMU2\APPWATCH.TAB);
```

Appendix B. Host NetView Related Files

B.1 CM/2 Configuration File

```
DEFINE_LOCAL_CP  FQ_CP_NAME(USIBMMK.MK333720 )
                  CP_ALIAS(WTR33372)
                  NAU_ADDRESS(INDEPENDENT_LU)
                  NODE_TYPE(EN)
                  NODE_ID(X'05D33372')
                  NW_FP_SUPPORT(NONE)
                  HOST_FP_SUPPORT(YES)
                  HOST_FP_LINK_NAME(HOST$1 )
                  MAX_COMP_LEVEL(NONE)
                  MAX_COMP_TOKENS(0);

DEFINE_LOGICAL_LINK LINK_NAME(HOST$1 )
                    FQ_ADJACENT_CP_NAME(USIBMMK.MK34      )
                    ADJACENT_NODE_TYPE(LEN)
                    DLC_NAME(IBMTRNET)
                    ADAPTER_NUMBER(0)
                    DESTINATION_ADDRESS(X'40003000000104')
                    ETHERNET_FORMAT(NO)
                    CP_SESSION_SUPPORT(NO)
                    SOLICIT_SSCP_SESSION(YES)
                    NODE_ID(X'05D33372')
                    ACTIVATE_AT_STARTUP(YES)
                    USE_PUNAME_AS_CPNAME(NO)
                    LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
                    LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                    MAX_ACTIVATION_ATTEMPTS(USE_ADAPTER_DEFINITION)
                    EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                    COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                    COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                    SECURITY(USE_ADAPTER_DEFINITION)
                    PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                    USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                    USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                    USER_DEFINED_3(USE_ADAPTER_DEFINITION);

DEFINE_PARTNER_LU  FQ_PARTNER_LU_NAME(USIBMRA.RAPAN      )
                  PARTNER_LU_ALIAS(NETVIEW)
                  PARTNER_LU_UNINTERPRETED_NAME(RAPAN      )
                  MAX_MC_LL_SEND_SIZE(32767)
                  CONV_SECURITY_VERIFICATION(NO)
                  PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU_LOCATION  FQ_PARTNER_LU_NAME(USIBMRA.RAPAN      )
                           WILDCARD_ENTRY(NO)
                           FQ_OWNING_CP_NAME(USIBMMK.MK34      )
                           LOCAL_NODE_NN_SERVER(NO);

DEFINE_DEFAULTS  IMPLICIT_INBOUND_PLU_SUPPORT(YES)
                 DEFAULT_MODE_NAME(BLANK)
                 MAX_MC_LL_SEND_SIZE(32767)
                 DIRECTORY_FOR_INBOUND_ATTACHES(*)
                 DEFAULT_TP_OPERATION(NONQUEUED_AM_STARTED)
                 DEFAULT_TP_PROGRAM_TYPE(BACKGROUND)
                 DEFAULT_TP_CONV_SECURITY_RQD(NO)
                 MAX_HELD_ALERTS(10);

DEFINE_REMOTE_FOCAL_POINT  SNA_DEFINED_MS_CATEGORY(X'23',031)
                          DESCRIPTION(ALERT CATEGORY)
                          FQ_PRIMARY_FP_NAME(USIBMRA.RAPAN      );

START_ATTACH_MANAGER;
```

B.2 Host VTAM Definitions

```
*****
*
*          VTAM SWITCHED MAJOR NODE FOR ITSC OFFICES          *
*
*****
          VBUILD MAXGRP=1,
              MAXNO=1,
              TYPE=SWNET
*
* MK33372
*
MK33372  PU      ADDR=13,
                  IDBLK=05D,
                  IDNUM=33372,
                  DISCNT=NO,
                  ISTATUS=ACTIVE,
                  MAXDATA=1033,
                  MAXPATH=4,
                  PACING=0,
                  PUTYPE=2,
                  DLOGMOD=D4C3290,
                  MODETAB=RGWBINDS,
                  USSTAB=USSTAB,
                  VPACING=0
*
MK333720 LU      LOCADDR=0,DLOGMOD=DSIL6MOD
MK33372A LU      LOCADDR=2
MK33372B LU      LOCADDR=3
MK33372C LU      LOCADDR=4
MK33372D LU      LOCADDR=5
*
```

Appendix C. Configuration Files for Our NV2MGR1 Machine

C.1 CONFIG.SYS File

```
IFS=C:\OS2\HPFS.IFS /CACHE:2048 /CRECL:32 /AUTOCHECK:CDEF
SET SQLNETB=16
PROTSHELL=C:\OS2\PMSHELL.EXE
SET USER_INI=C:\OS2\OS2.INI
SET SYSTEM_INI=C:\OS2\OS2SYS.INI
SET OS2_SHELL=C:\OS2\CMD.EXE
SET AUTOSTART=PROGRAMS,TASKLIST,FOLDERS,CONNECTIONS
SET RESTARTOBJECTS=STARTUPFOLDERONLY
SET RUNWORKPLACE=C:\OS2\PMSHELL.EXE
SET COMSPEC=C:\OS2\CMD.EXE
LIBPATH=D:\ANV2\READER;D:\ANV2\DLL;D:\ANV2\DLL\AGENT;C:\IBMC\DLL;D:\SQLLIB\DLL;.;C:\USERDLLS;C
:\OS2\DLL;C:\IBMLAN\NETLIB;
C:-MUGLIB-DLL;C:-OS2-APPS-DLL;C:-CMLIB-DLL;
C:-OS2-MDOS;C:-;D:-TCPIP-DLL;C:-NETWARE;C:-NETWARE-NLS-ENGLISH;
L:-OS2;P:-OS2;D:-TCPIP-PMX-DLL;D:-LMU2;--WTRDC-LANDLLS;
SET
PATH=D:\ANV2\READER;D:\ANV2\BIN\AGENT;D:\ANV2\BIN;D:\ANV2\ETC;D:\SQLLIB;C:\CMDS;C:\OS2UTIL
S;C:\OS2;C:\IBMLAN\NETPROG;
C:-MUGLIB;C:-CMLIB;C:-OS2-SYSTEM;
C:-OS2-APPS;C:-OS2-MDOS-WINOS2;C:-OS2-INSTALL;C:-OS2-MDOS;C:-;
D:-TCPIP-BIN;C:-NETWARE;L:-OS2;P:-OS2;D:-TCPIP-PMX-BIN;D:-LMU2;
SET
DPATH=D:\ANV2\ETC\LRF;D:\ANV2\HELP;D:\ANV2\ETC;C:\IBMC\;D:\TCPIP\BIN;D:\SQLLIB;C:\OS2UTIL
S;C:\OS2;C:\IBMLAN\NETPROG;C:\IBMLAN;
C:-MUGLIB;C:-CMLIB;C:-OS2-SYSTEM;C:-OS2-APPS;
C:-OS2-MDOS-WINOS2;C:-OS2-INSTALL;C:-OS2-BITMAP;C:-OS2-MDOS;
C:-;C:-NETWARE;C:-NETWARE-NLS-ENGLISH;L:-NLS;P:-NLS;D:-LMU2;
SET PROMPT=$e-7mOS2$e-0m $P$G
SET
HELP=D:\ANV2\READER;D:\ANV2\HELP;D:\SQLLIB;C:\OS2\HELP;C:\HELPLIB;C:\OS2\HELP\TUTORIAL;C:\
CMLIB;D:\TCPIP\HELP;
C:-NETWARE-NLS-ENGLISH;D:-TCPIP-PMX-HELP;D:-LMU2;
SET GLOSSARY=C:\OS2\HELP\GLOSS;
SET IPF_KEYS=SBCS
PRIORITY_DISK_IO=YES
FILES=30
DEVICE=C:\IBMC\PROTOCOL\LANPDD.OS2
DEVICE=C:\IBMC\PROTOCOL\LANVDD.OS2
DEVICE=C:\IBMC\LANMSGDD.OS2 /I:C:\IBMC
DEVICE=C:\IBMC\PROTMAN.OS2 /I:C:\IBMC
DEVICE=C:\OS2\TESTCFG.SYS
DEVICE=C:\OS2\DOS.SYS
DEVICE=C:\OS2\PMDD.SYS
BUFFERS=30
IOPL=YES
MAXWAIT=3
MEMMAN=SWAP,PROTECT
SWAPPATH=D:\ 2048 12288
BREAK=OFF
THREADS=512
PRINTMONBUFSIZE=134,134,134
COUNTRY=001,C:\OS2\SYSTEM\COUNTRY.SYS
SET KEYS=ON
BASEDEV=PRINT02.SYS
BASEDEV=IBM2FLPY.ADD
BASEDEV=IBM2SCSI.ADD /LED
BASEDEV=OS2DASD.DMD
```

```

SET
BOOKSHELF=D:\SQLLIB\BOOK;C:\IBMLAN\BOOK;C:\OS2\BOOK;C:\INFLIB;C:\CMLIB\BOOK;d:\TCP\DOC;D:\TCP\
P\pmx\DOC;
SET  EPMPATH=C:\OS2\APPS;
PROTECTONLY=NO
SHELL=C:\OS2\MDOS\COMMAND.COM C:\OS2\MDOS
FCBS=16,8
RMSIZE=640
rem  DEVICE=C:\NETWARE\VIPX.SYS
rem  DEVICE=C:\NETWARE\VSHELL.SYS PRIVATE
DEVICE=C:\OS2\MDOS\VEMM.SYS
DOS=LOW,NOUMB
DEVICE=C:\OS2\MDOS\VXMS.SYS /UMB
DEVICE=C:\OS2\MDOS\VDPMI.SYS
DEVICE=C:\OS2\MDOS\VDPX.SYS
DEVICE=C:\OS2\MDOS\VCDROM.SYS
DEVICE=C:\OS2\MDOS\VWIN.SYS
DEVICE=C:\OS2\MDOS\VMOUSE.SYS
DEVICE=C:\OS2\POINTDD.SYS
DEVICE=C:\OS2\MOUSE.SYS
DEVICE=C:\OS2\COM.SYS
DEVICE=C:\OS2\MDOS\VCOM.SYS
CODEPAGE=437,850
DEVINFO=KBD,US,C:\OS2\KEYBOARD.DCP
BASEDEV=XGA.SYS
DEVICE=C:\OS2\XGARINGO.SYS
DEVINFO=SCR,VGA,C:\OS2\VIOTBL.DCP
SET  VIDEO_DEVICES=VIO_XGA
SET  VIO_XGA=DEVICE(BVHVGA,BVHXGA)
DEVICE=C:\OS2\MDOS\VVGA.SYS
DEVICE=C:\OS2\MDOS\VXGA.SYS
RUN=C:\IBMCOM\PROTOCOL\NETBIND.EXE
RUN=C:\IBMCOM\LANMSGEX.EXE
DEVICE=C:\IBMCOM\PROTOCOL\NETBEUI.OS2
DEVICE=C:\IBMLAN\NETPROG\RDRHELP.200
REM --- NetWare Requester statements BEGIN ---
SET  NWLANGUAGE=ENGLISH
DEVICE=C:\NETWARE\LSL.SYS
RUN=C:\NETWARE\DDAEMON.EXE
DEVICE=C:\IBMCOM\PROTOCOL\ODI2NDI.OS2

REM -- ODI-Driver Files BEGIN --
REM  DEVICE=C:\NETWARE\TOKEN.SYS
REM -- ODI-Driver Files END --
DEVICE=C:\NETWARE\ROUTE.SYS
DEVICE=C:\NETWARE\IPX.SYS
DEVICE=C:\NETWARE\SPX.SYS
RUN=C:\NETWARE\SPDAEMON.EXE
DEVICE=C:\NETWARE\NMPIPE.SYS
DEVICE=C:\NETWARE\NPSEVER.SYS
RUN=C:\NETWARE\NPDAEMON.EXE
DEVICE=C:\NETWARE\NWREQ.SYS
IFS=C:\NETWARE\NWIFS.IFS
RUN=C:\NETWARE\NWDAEMON.EXE
DEVICE=C:\NETWARE\NETBIOS.SYS
RUN=C:\NETWARE\NBDAEMON.EXE
REM  DEVICE=C:\OS2\MDOS\LPTDD.SYS
REM --- NetWare Requester statements END ---
IFS=C:\IBMLAN\NETPROG\NETWKSTA.200 /I:C:\IBMLAN /N
DEVICE=C:\IBMCOM\PROTOCOL\NETBIOS.OS2
DEVICE=C:\IBMCOM\PROTOCOL\LANDD.OS2
DEVICE=C:\IBMCOM\PROTOCOL\LANDLLDD.OS2
DEVICE=C:\IBMCOM\MACS\IBMTOK.OS2
RUN=C:\IBMCOM\PROTOCOL\LANDLL.EXE
DEVICE=C:\CMLIB\ACSLANDD.SYS
DEVICE=C:\CMLIB\CMKFMDE.SYS
SET  CMPATH=C:\CMLIB
DEVICE=C:\OS2\EPWDD.SYS
RUN=C:\OS2\EPWDDR3.EXE
RUN=C:\OS2\EPW.EXE
RUN=C:\IBMLAN\NETPROG\LSDAEMON.EXE
RUN=C:\OS2\SYSTEM\LOGDAEM.EXE
RUN=C:\OS2\EPWROUT.EXE 1
DEVICE=C:\OS2\LOG.SYS
DEVICE=C:\IBMLAN\NETPROG\VNETAPI.OS2
RUN=C:\IBMLAN\NETPROG\VNRMINIT.EXE
SET  INCLUDE=D:\ANV2\TOOLKIT\INCLUDE;D:\SQLLIB;
SET  LIB=D:\ANV2\TOOLKIT\LIB;D:\SQLLIB;
SET  QRWDR=D
SET  QRWINST=D:\SQLLIB

```

```

SET ETC=d:\TCPIP\ETC
SET TMP=d:\TCPIP\TMP
SET READIBM=D:\ANV2\BOOKS;D:\TCPIP\DOC;D:\TCPIP\PMX\DOC;D:\LMU2;
SET HOSTNAME=nv2mgr1
RUN=d:\TCPIP\BIN\CNTRL.EXE
REM Replaced by AnyNet/2 Sockets over SNA *
REM DEVICE=C:\IBMCOM\PROTOCOL\INET.SYS
DEVICE=C:\IBMCOM\PROTOCOL\IFNDIS.SYS
SET DISPLAY=nv2mgr1:0
SET XFILES=D:\TCPIP\pmx\X11
SET NLSPATH=D:\ANV2\NLS\%N
SET SYSCONTACT=Mirek Iwachow
SET SYSLOCATION=Raleigh, bld 062, room L610
SET SXMODE_DEFAULT=SNACKETS
DEVICE=D:\TCPIP\BIN\SNACKETS.SYS
SET TELNET.PASSWORD.ID=cm5001r
DEVICE=D:\LMU2\LMUIPL.SYS
SET BOOKMGR=D:\LMU2;
SET USER=nv4os2
SET PASSWD=nv4os2
rem DEVICE=D:\THESEUS2\THESEUS2.SYS
SET NETVIEW_PATH=D:\ANV2
DEVICE=D:\ANV2\BIN\AGENT\THESEUS2.SYS
SET SNMPDIR=D:\ANV2\ETC
SET NMS=D:\ANV2\ETC\DBASE
SET BTRPARMS=/M:48 /P:4096 /U:2 /F:100 /T:D:\ANV2\ETC\DBASE\BTRIEVE.TRN
SET NMSADMIN=IPXSEARCH

```

C.2 PROTOCOL.INI File

PROT_MAN

DRIVERNAME = PROTMAN\$

IBMLXCFG

landd_nif = landd.nif
netbeui_nif = netbeui.nif
odi2ndi_nif = odi2ndi.nif
tcpip_nif = tcpip.nif
IBMTOK_nif = ibmtok.nif

landd_nif

DriverName = LANDD\$
Bindings = IBMTOK_nif
NETADDRESS = "T400000033372"
ETHERAND_TYPE = "I"
SYSTEM_KEY = 0x0
OPEN_OPTIONS = 0x2000
TRACE = 0x0
LINKS = 80
MAX_SAPS = 8
MAX_G_SAPS = 0
USERS = 7
T1_TICK_G1 = 255
T1_TICK_G1 = 15
T2_TICK_G1 = 3
T1_TICK_G2 = 255
T1_TICK_G2 = 25
T2_TICK_G2 = 10
IPACKETS = 250
UIPACKETS = 100
MAXTRANSMITS = 6
MINTRANSMITS = 2
TCBS = 64
GDTS = 30
ELEMENTS = 800

netbeui_nif

DriverName = netbeui\$
Bindings = IBMTOK_nif
NETADDRESS = "T400000033372"
ETHERAND_TYPE = "I"
USEADDRREV = "YES"
OS2TRACEMASK = 0x0
SESSIONS = 128
NCBS = 128
NAMES = 80
SELECTORS = 5
USEMAXDATAGRAM = "NO"
ADAPTRATE = 1000
WINDOWERRORS = 0
MAXDATARCV = 4168
TI = 30000
T1 = 500

```
T2 = 200
MAXIN = 1
MAXOUT = 1
NETBIOS_TIMEOUT = 500
NETBIOS_RETRIES = 8
NAMECACHE = 0
PIGGYBACKACKS = 1
DATAGRAM_PACKETS = 2
PACKETS = 350
LOOP_PACKETS = 1
PIPELINE = 5
MAX_TRANSMITS = 6
MIN_TRANSMITS = 2
DLC_RETRIES = 5
FC_PRIORITY = 5
NET_FLAGS = 0x0
```

odi2ndi_nif

```
DriverName = odi2ndi$
Bindings = IBMTOK_nif
NETADDRESS = "T400000033372"
TOKEN_RING = "yes"
TOKEN_RING_SNAP = "no"
ETHERNET_802.3 = "no"
ETHERNET_802.2 = "no"
ETHERNET_II = "no"
ETHERNET_SNAP = "no"
TRACE = 0x0
```

tcipip_nif

```
DriverName = TCPIP$
Bindings = IBMTOK_nif
```

IBMTOK_nif

```
DriverName = IBMTOK$
ADAPTER = "PRIMARY"
NETADDRESS = "400000033372"
MAX_TRANSMITS = 6
RECVBUFS = 2
RECVBUFSIZE = 256
XMITBUFS = 1
```

Index

A

- agents
 - DOS and DOS/Windows 145—148
 - LAN Server and Requester 129—143
 - NetWare 149—153
 - OS/2 95—127
 - SIA (System Information Agent) 95
- alert vectors 223
- alerts, generic
- AnyNet/2 Sockets over SNA
 - description 2
- API 2
- APPC communication 196
- APPC environment 189
- application programming interface 2
- automating actions from NetView from MVS
 - alert vectors 223
 - automation table 223
 - description 223—261
 - MSU 223
 - NMVT 223
 - OS/2 LAN Requester 223
- automation table 223

B

- building MIB applications 66

C

- collecting and displaying MIB variable types 240
- Communications Manager/2
 - APPC environment 189
 - connecting to MVS/ESA 189—236
 - setup 189
- configuration
 - LMU 157
 - sample files 293
- CPU utilization 70

D

- Data Collector 75, 240, 249
- DOS agent
 - description 145
- DOS Windows agent
 - description 147
- DSI6DST task 204

E

- environment, ITSO lab
 - description 9—44
 - hardware 9

- environment, ITSO lab (*continued*)
 - IBM TCP/IP for OS/2 and Database Manager 2/2
 - installation 20
 - installing the NetWare Requester 12
 - software 9

F

- files, configuration 293

G

- gathering Vital Product Data (VPD) 182
- Generic-Trap, SNMP 215
- graphing CPU utilization 70
- graphing MIB variables 64

H

- host configuration 99
- host resources MIB 99
- host-related files
 - NetView files 291—292

I

- Internet Protocol
 - description 1
- IP
 - See Internet Protocol
- IPX
 - description 1
 - in discovery process 45

L

- LAN Management Utility
 - See LMU (LAN Management Utility)
- LAN requester agent
- LAN Server 3.0 97
- LAN server agent
- Lexmark 4039 SNMP printer
 - description 249—259
 - enterprise-specific MIB 249
- LMU (LAN Management Utility)
 - configuration 157
 - gathering vital product data 182
 - installation and customization 155
 - integrating LMU alerts into NetView for OS/2 169
 - interactions 155—187
 - MIB 106
 - remote commands from NetView for OS/2 168
 - SNMP proxy agent 106
 - startup 166

LMU MIB 5, 106
LMU/2-related files
LMUCMD 220
LU 6.2 communications 192
LU6.2 sessions 205

M

Management Information Base)
 See MIB (Management Information Base)
MIB (Management Information Base)
 Data Collector 240
 description 3
 enterprise-specific 249
 enterprise-specific MIB 3
 host resources 99
 standard MIB 3
MIB applications
 building applications 66
 description 66
 graphing CPU utilization 70
MIB browser
 collecting and displaying MIB variable types 240
 description 62
 graphing MIB variables 64
 managing Lexmark printer 249
MIB data collector 75
MIB instance 77
MIB loader
 description 60
MSU 223
MVS
 DSI6DST task 204
 NetView for OS/2 connecting to 189–236
 RUNCMD 219

N

NetBIOS
 description 1
 in discovery process 45
 LMU example 179
netbiosdiscovery 45
NetView
 host-related files 291–292
NetView for OS/2
 discovery process 45
 Event Automation and the Event Displayer 83
 functional overview 45–93
 Host Connectivity 93
 MIB applications 66
 MIB browser 62
 MIB Data Collector 75
 MIB loader 60
 netbiosdiscovery 45
 netwdiscovery 45
 seed file, use of 45
 tcpipdiscovery 45

NetWare agent
 servers, configuration 165
 servers, installation 156
netwdiscovery 45
network protocols
 AnyNet/2 2
 description 1–2
 Internet Protocol 1
 IPX 1
 NetBIOS 1
NMVT 223
NPDA command 211

O

OS/2 LAN Requestor 179

P

polling interval 3, 78
proxy agent, SNMP 180

R

REXX
 SNMP commands 237
 SNMPGET 237
 SNMPNEXT 237
 SNMPSET 237
 SNMPTRAP 237
 SNMPWALK 237
 use with SNMP commands 237–247
RUNCMD 219

S

seed file 45
service point name 219
SIA
 See System Information Agent (SIA)
SNMP (Simple Network Management Protocol)
 agent functions 4
 description 2–8
 devices 189
 events and traps 5
 examples of monitoring 95
 Generic-Trap Number Authentication Failure 215
 LMU MIB 5
 NetView for OS/2 SNMP commands
 SNMPGET 6
 SNMPNEXT 7
 SNMPSET 7
 SNMPTRAP 7
 SNMPWALK 8
 polling interval 3
 proxy agent 180
 proxy agent functions 4
 proxy agent, LMU 106
 SNMP commands
 GET 6

SNMP (Simple Network Management Protocol)

(continued)

SNMP commands *(continued)*

GET NEXT 6

GET RESPONSE 6

SET 6

SNMPGET 237

SNMPNEXT 237

SNMPSET 237

SNMPTRAP 237

SNMPWALK 237

TRAP 6

SNMP VI 4

statistical data 4

trap 4

sname 219

System Information Agent (SIA)

description 117–129

monitoring the MIB, examples 95

T

TCP/IP

in discovery process 45

tcpipdiscovery 45

V

Vital Product Data

gathering, examples 182

LMU database 115

VPD

See Vital Product Data

**International Technical Support Organization
NetView for OS/2 as an SNMP Manager
December 1994**

Publication No. GG24-4412-00

Your feedback is very important to help us maintain the quality of ITSO Bulletins. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

Please rate on a scale of 1 to 5 the subjects below.

(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Organization of the book _____
Accuracy of the information _____
Relevance of the information _____
Completeness of the information _____
Value of illustrations _____

Grammar/punctuation/spelling _____
Ease of reading and understanding _____
Ease of finding information _____
Level of technical detail _____
Print quality _____

Please answer the following questions:

- a) If you are an employee of IBM or its subsidiaries:
Do you provide billable services for 20% or more of your time? Yes_____ No_____
Are you in a Services Organization? Yes_____ No_____
- b) Are you working in the USA? Yes_____ No_____
- c) Was the Bulletin published in time for your needs? Yes_____ No_____
- d) Did this Bulletin meet your needs? Yes_____ No_____
- If no, please explain:

What other topics would you like to see in this Bulletin?

What other Technical Bulletins would you like to see published?

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



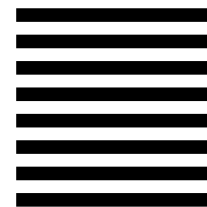
BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Organization
Department 545, Building 657
P.O. BOX 12195
RESEARCH TRIANGLE PARK NC
USA 27709-2195

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



Fold and Tape

Please do not staple

Fold and Tape



Printed in U.S.A.

GG24-4412-00

